

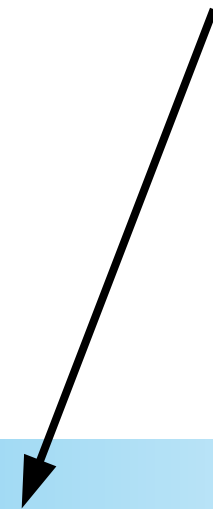
# Home Automation with IoT



**A POSTERIORI**  
Play · Experience · Learn

# Before we start...

- We believe in open access to knowledge
- All our slides are shared online for free
- You can print it, share it, modify it, use it to run your own IoT companies...
- This current set of slides can be found here



# About Us - A Posteriori



## YONI

Spent 15 years developing software in the financial service industry, now developing the next generation of Makers and Coders.

## CORT

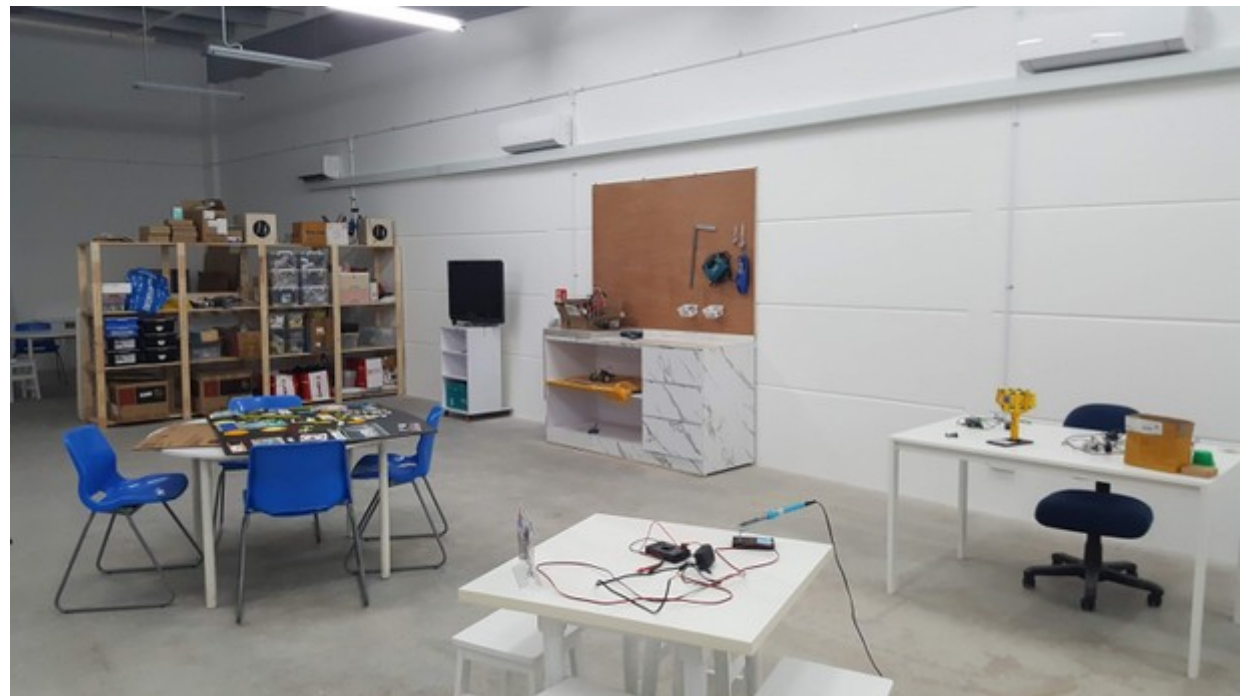
Ex-Navy engineer managing big engines, powerful generators, and easily choked toilets. Codes and builds stuff because he's too cheap to buy



# Our Company

## Our Makerspace

- Suitable for electronics, 3D printing, woodwork, coding and tinkering
- Free for public use
- Located in MEGA @ Woodlands



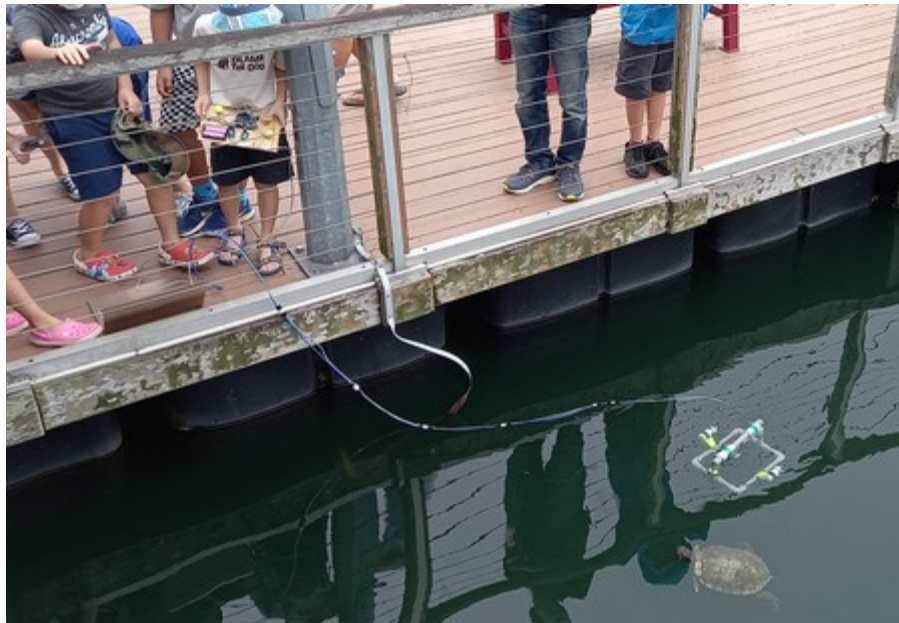
# Our Programmes

- Competition Programs
  - Our team, Galaxy Raiders, participates in various national events, collecting trophies and medals...



# Our Programmes

- Holiday Camps
  - Robotics, Apps creation, Coding, 3D Designs, Science and Research



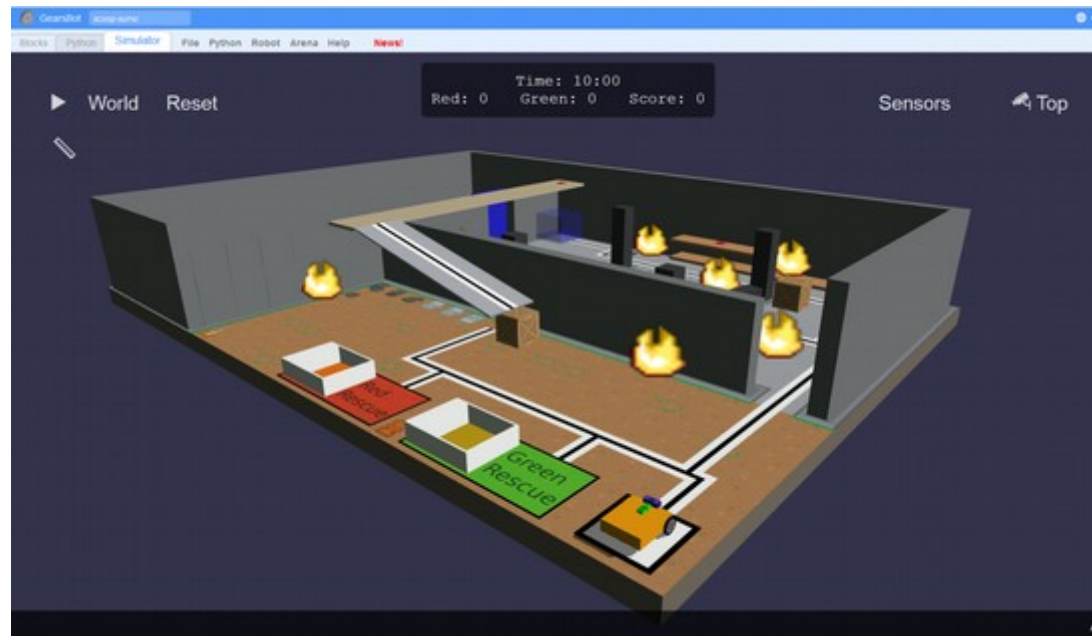
# Our Programmes

- Regular Sessions

- Virtual Coding/Robotics @ Home (over Zoom)
- Interest-based, guided exploration

## GearsBot

Cort's  
Circuit-Breaker  
brain-child.



# What is the Internet-of-Things (IoT)?

**A POSTERIORI**  
Play · Experience · Learn



# What is IoT?



Control lights through phone...



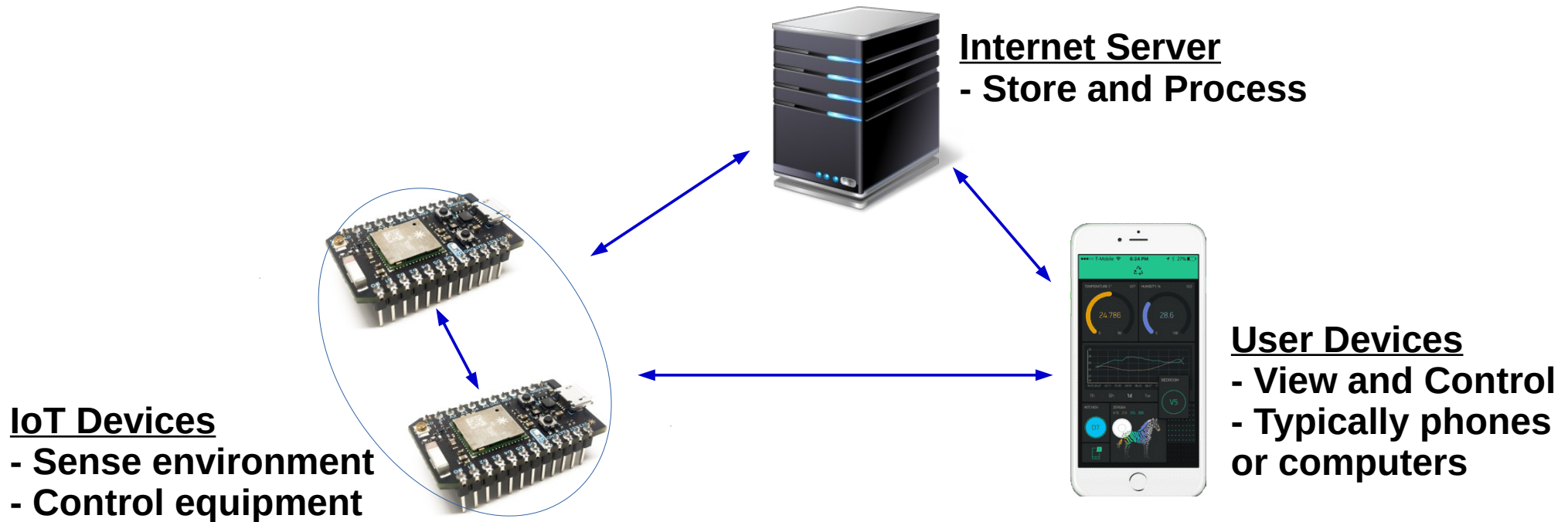
Turn on aircon before reaching home...



Toast your bread remotely (...we don't know why either)

- Make existing or new devices more useful by connecting them to the internet (eg. lights, aircon, door locks, burglar alarm)

# How Does IoT Work?



# IoT in Industry



Notify vendor to top-up machine when empty

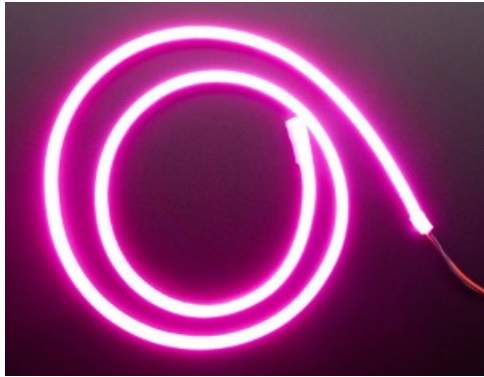


Track water and power usage

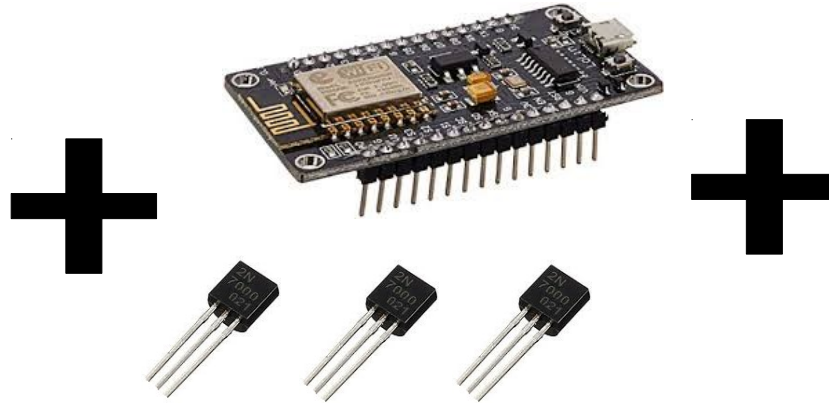


Arrange for garbage collection when bin is full

# IoT Can Be Cheap & Easy!



**LED Strips**  
(\$15 with power supply)



**Controller and transistors (\$10)**

```
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char auth[] = "xxx";

char ssid[] = "HomeWifi";
char pass[] = "yyy";

void setup()
{
  Blynk.begin(auth, ssid, pass);
}

void loop()
{
  Blynk.run();
}
```

**13 lines of code**  
**(5 mins of your time)**



**Phone controllable  
mood lighting**



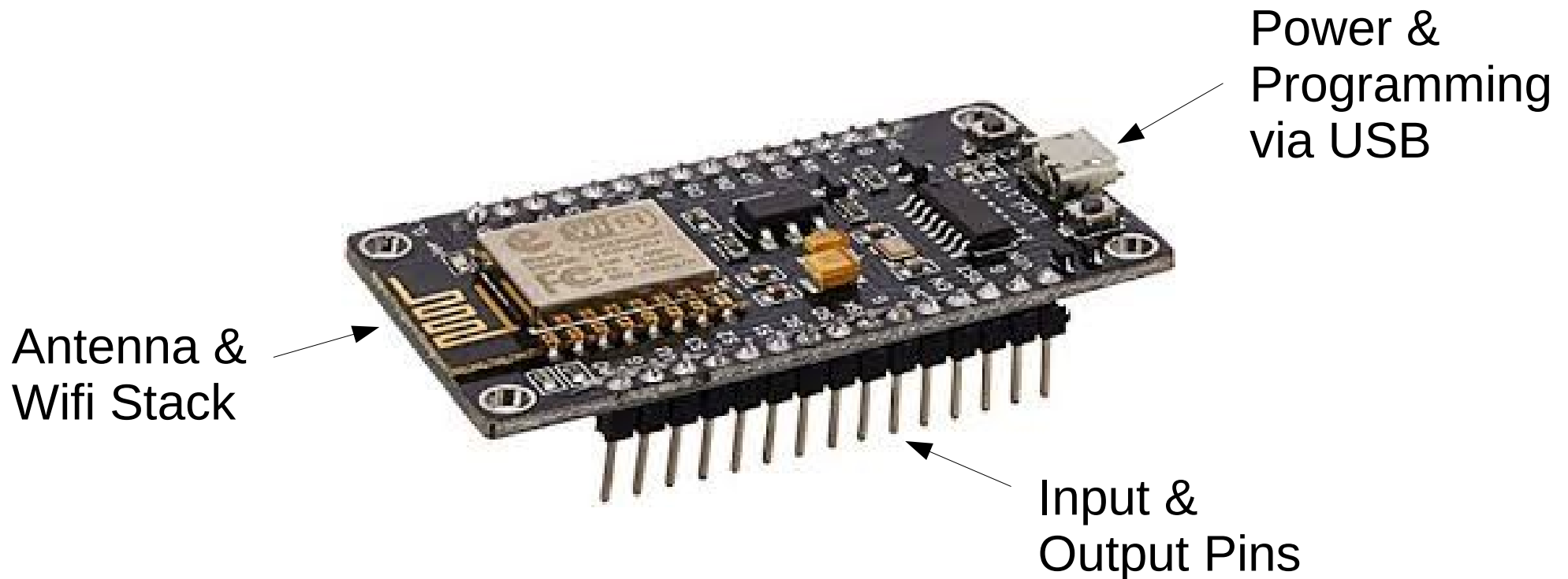
# The ESP8266 (IoT Device)



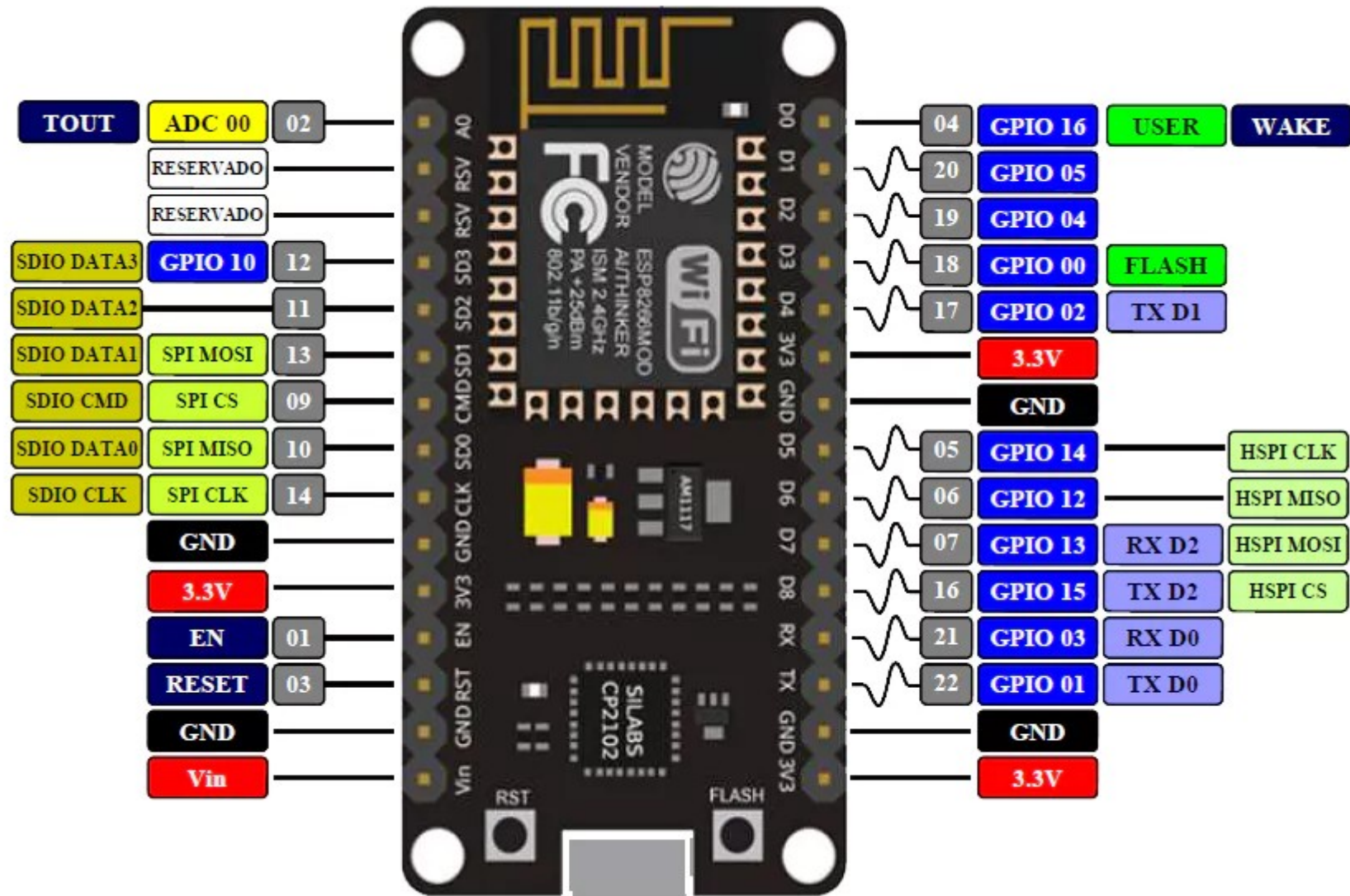
**A POSTERIORI**  
Play · Experience · Learn

# What is the ESP8266?

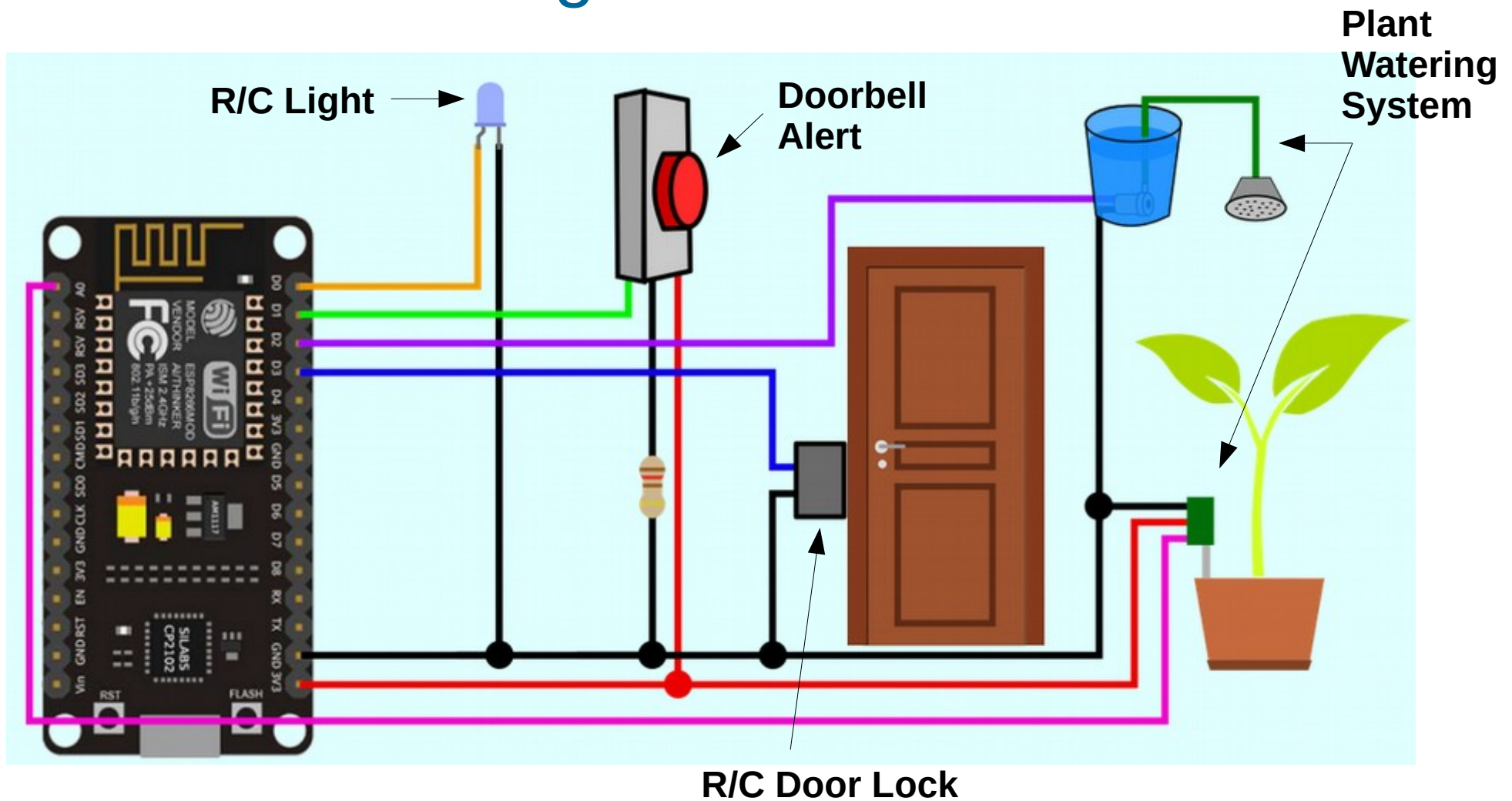
- Microcontroller (similar to Arduino and micro:bit)
- Built-in WiFi



# Pinouts



# Introducing Our Virtual Home



<http://a9i.sg/iot>

**A POSTERIORI**  
Play · Experience · Learn

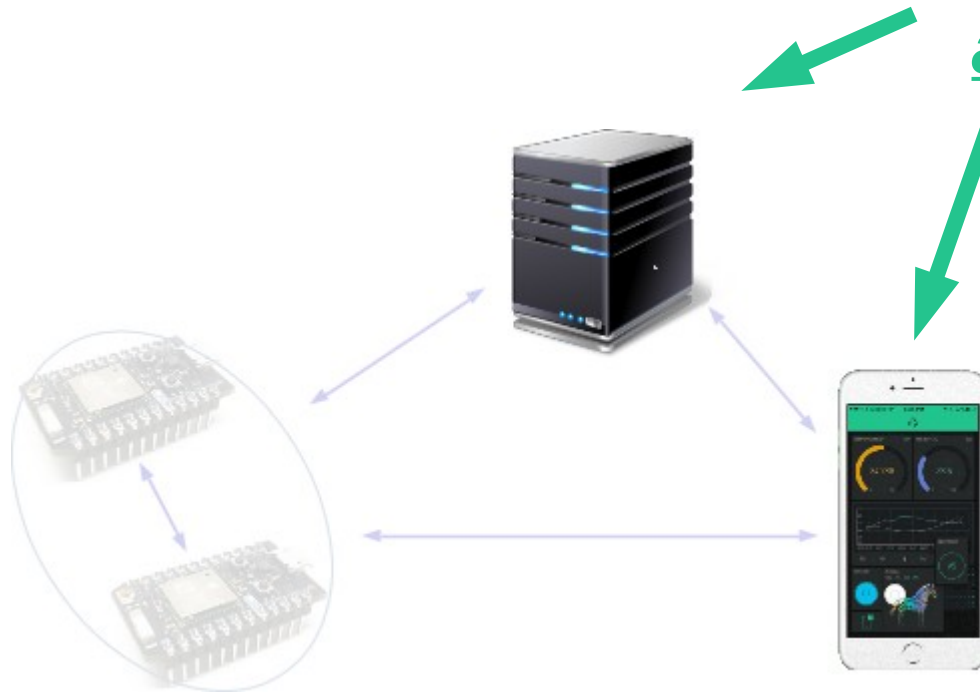


# Easy IoT With Blynk

**A POSTERIORI**  
Play · Experience · Learn



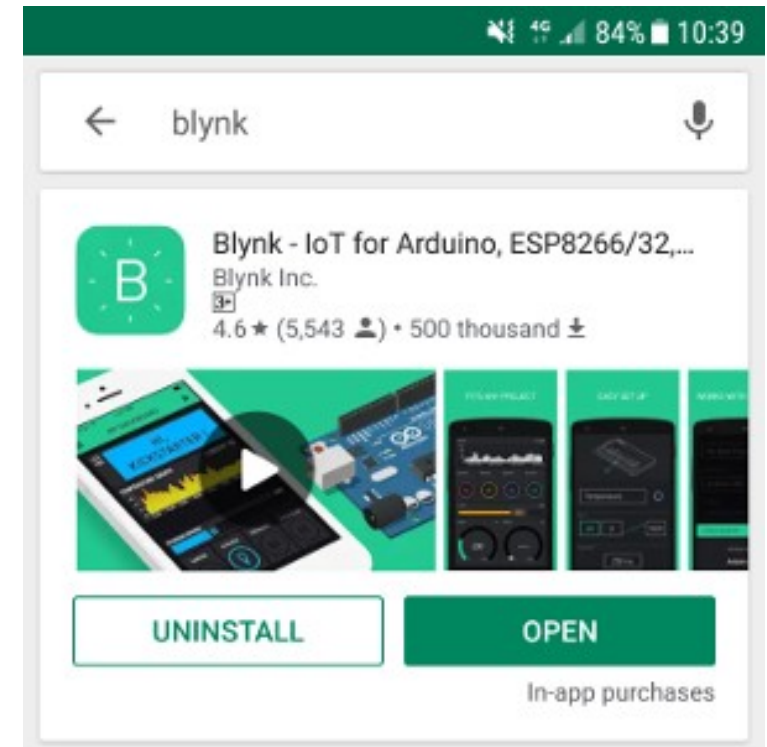
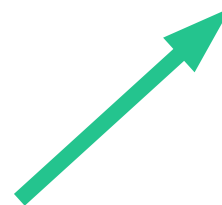
Blynk provides the server and the mobile app for your IoT device.



**A POSTERIORI**  
Play · Experience · Learn

# Blynk

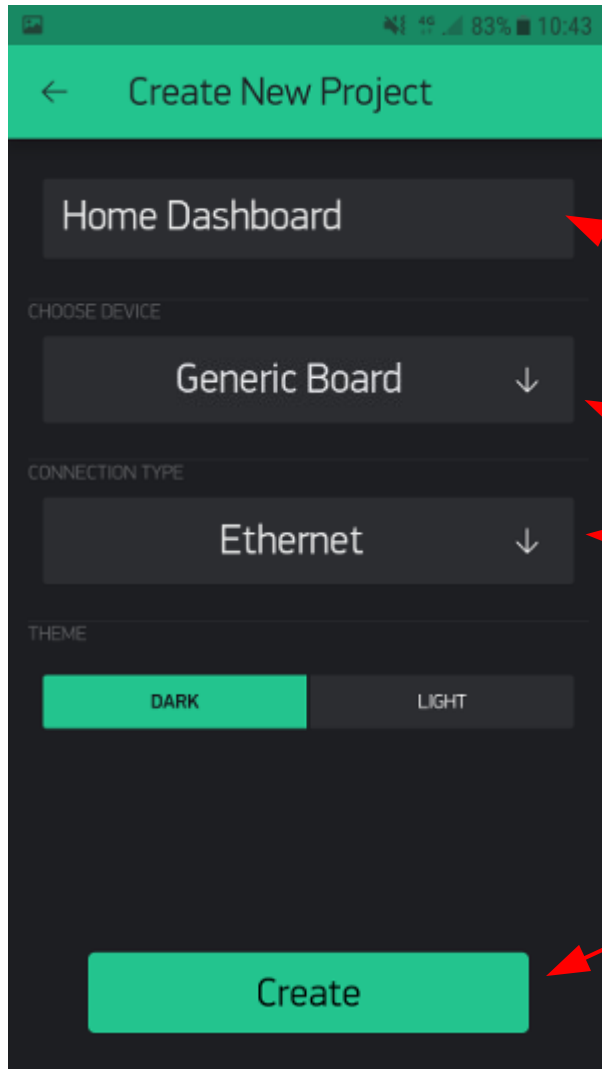
- Install Blynk
- Available for iOS or Android
- Register and Login



# Challenge 1: Connect your IoT Device

**A POSTERIORI**  
Play · Experience · Learn

# Challenge 1 : Connect IoT Device

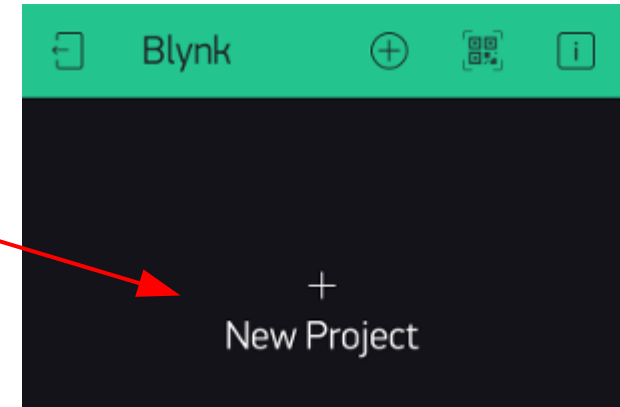


Create a new project

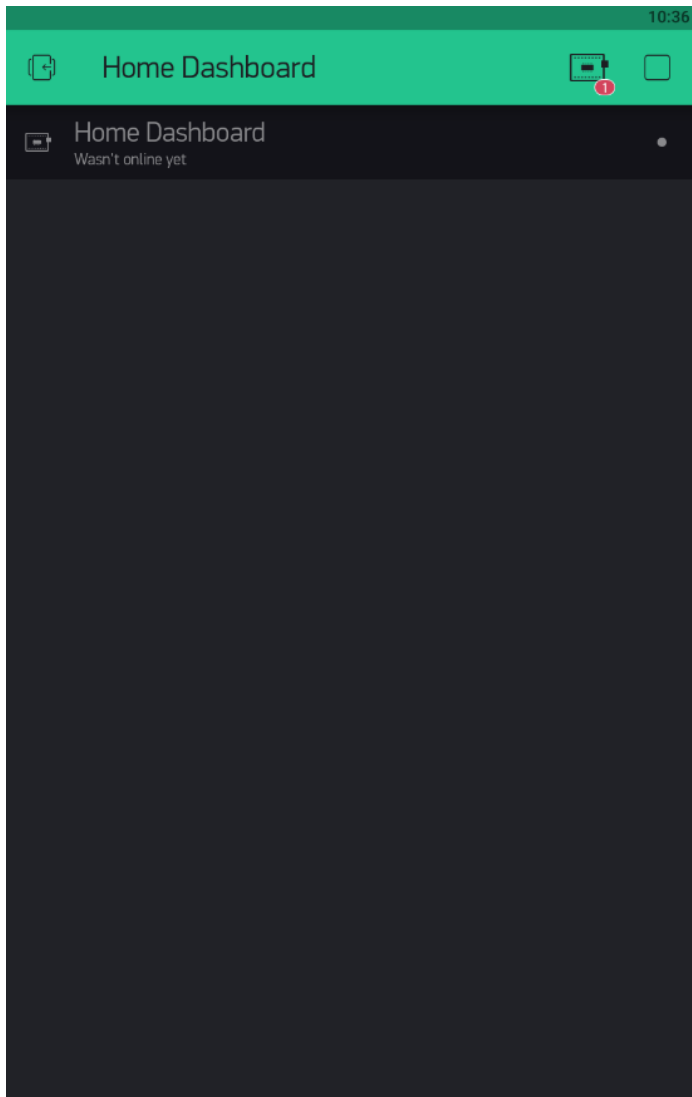
Choose Name

Set "Device" to "Generic Board" and "Connection Type" to "Ethernet"

Tap "Create" when you're done



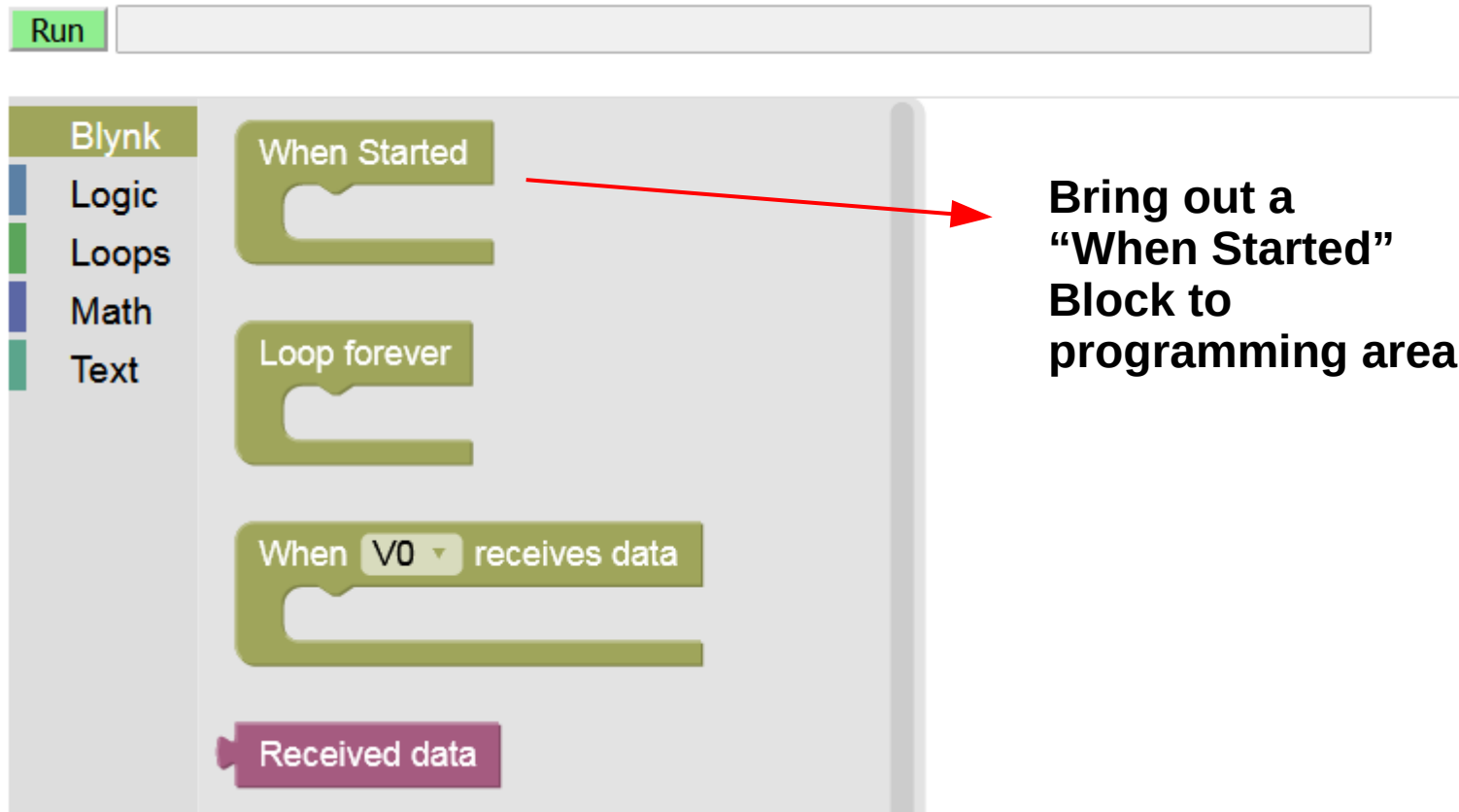
# Challenge 1 : Connect IoT Device



**Press Play**

**Click on Device icon  
Should say: "Wasn't online yet"**

# Challenge 1 : Connect IoT Device

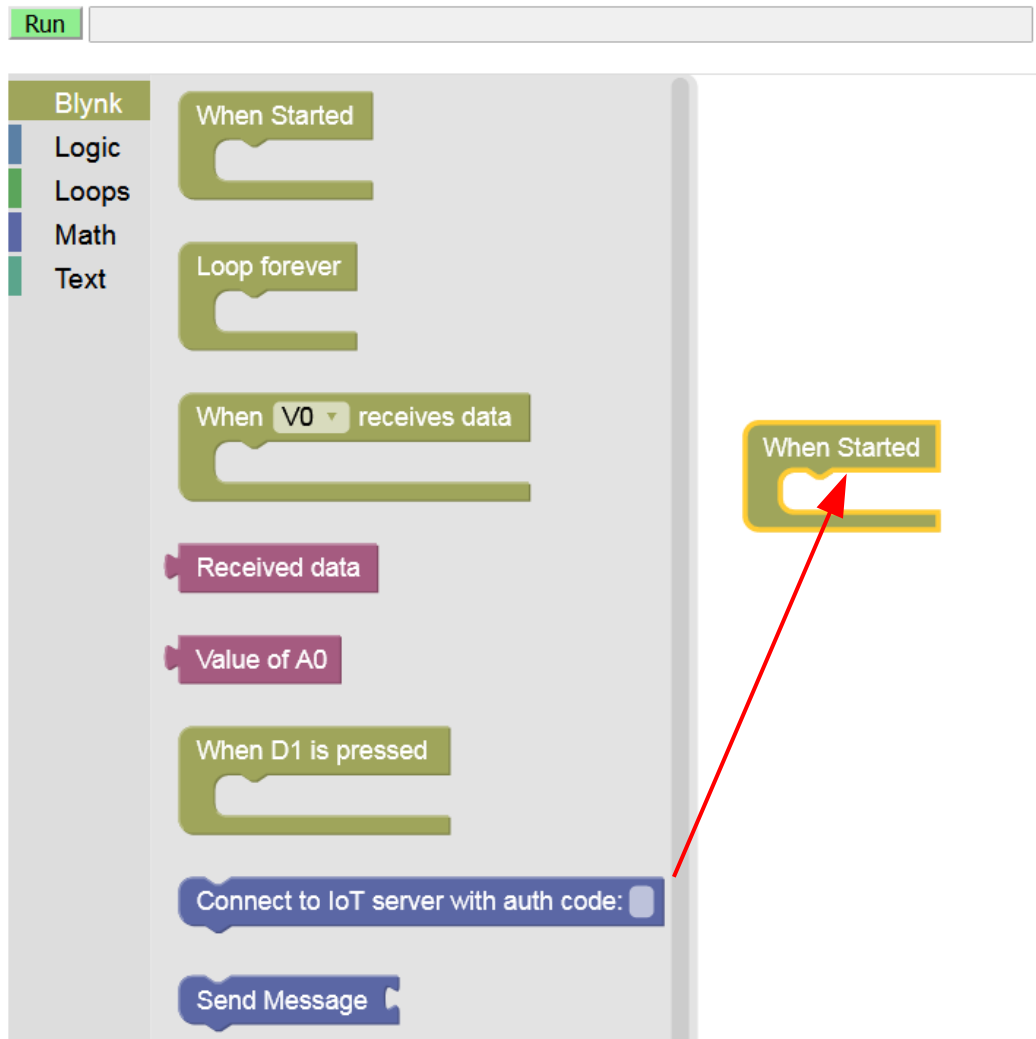


The image shows a screenshot of the Blynk IDE interface. At the top, there is a 'Run' button. Below it, a sidebar on the left contains a category menu with 'Blynk' selected, and sub-categories: 'Logic', 'Loops', 'Math', and 'Text'. The main workspace shows three blocks: 'When Started', 'Loop forever', and 'When V0 receives data'. A red arrow points from the 'When Started' block in the workspace to the right, where text reads: 'Bring out a "When Started" Block to programming area'. Below the workspace, there is a 'Received data' block.

**Bring out a  
"When Started"  
Block to  
programming area**

# Challenge 1 : Connect IoT Device

Run



The screenshot shows the Blynk IDE interface. On the left, there is a sidebar with categories: Blynk, Logic, Loops, Math, and Text. The main workspace contains several blocks: a 'When Started' block, a 'Loop forever' block, a 'When V0 receives data' block, a 'Received data' block, a 'Value of A0' block, a 'When D1 is pressed' block, a 'Connect to IoT server with auth code:' block, and a 'Send Message' block. A red arrow points from the 'Connect to IoT server with auth code:' block to a 'When Started' block in a separate window.

Place a “Connect to IoT Server” block inside the When Started block.



# Challenge 1 : Connect IoT Device

**Blynk** <dispatcher@blynk.io> [Unsubscribe](#)  
to cort ▾  
Auth Token : 6540315149b840cc857f771d47db5504

Check your email for  
the Auth Token...

...and put it in here.

Press "Run"  
when ready

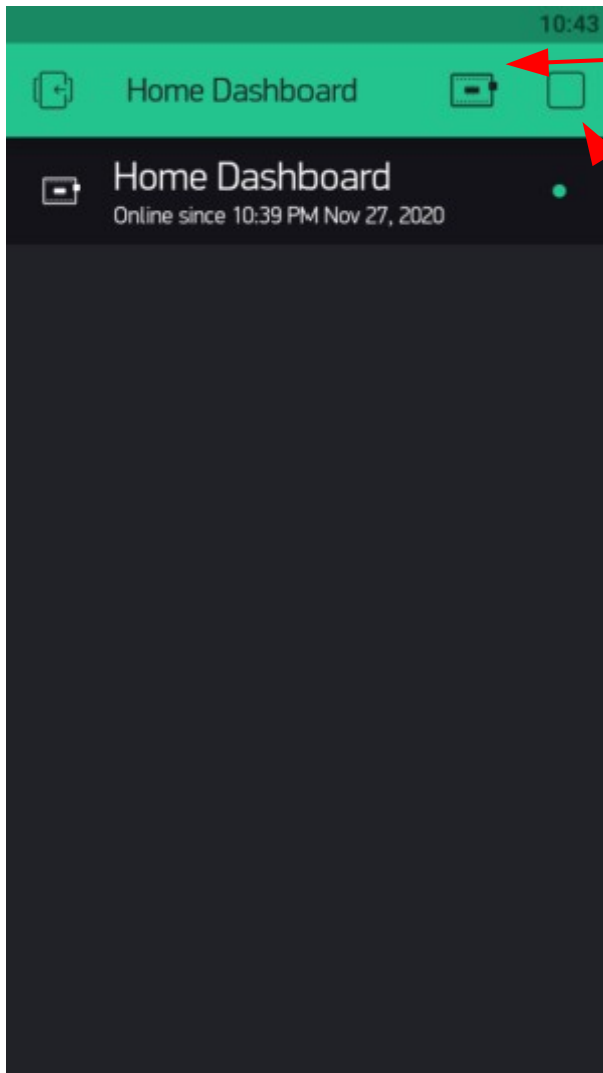
Run

Blynk  
Logic  
Loops  
Math  
Text

When Started

Connect to IoT server with auth code: Wu0tCCEBwGAKkGdbBJPuXf7ShE09\_rOq

# Challenge 1 : Connect IoT Device



**Warning icon should turn off**

**Click on Device icon  
Should say: "Online since..."**

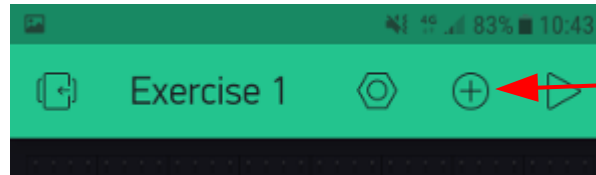
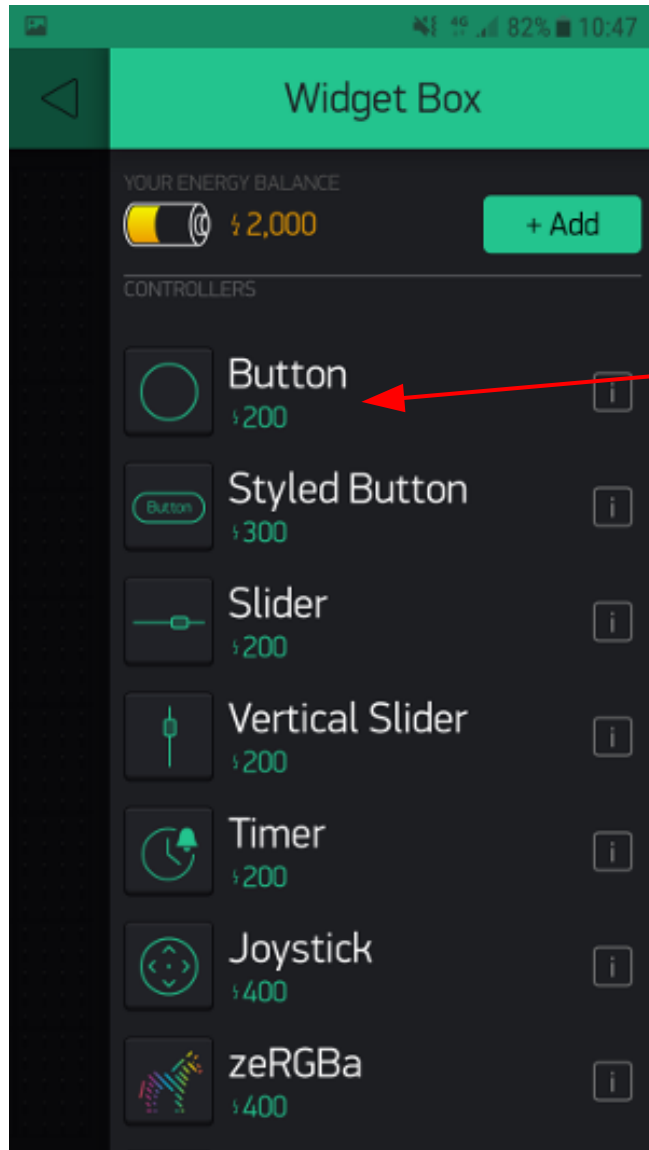
**CONGRATULATIONS!  
You've Connected your first IoT Device.**

**Now Click the Stop Button, so we can  
add some UI to our Dashboard.**

# Challenge 2: Add a Remote Light Switch

**A POSTERIORI**  
Play · Experience · Learn

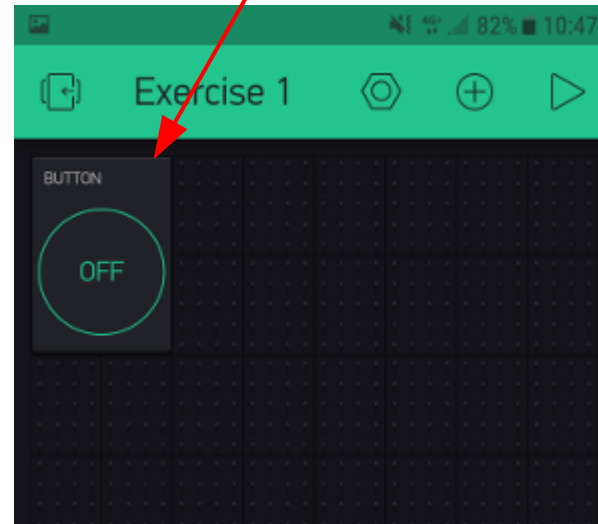
# Challenge 2 : Add Light Switch



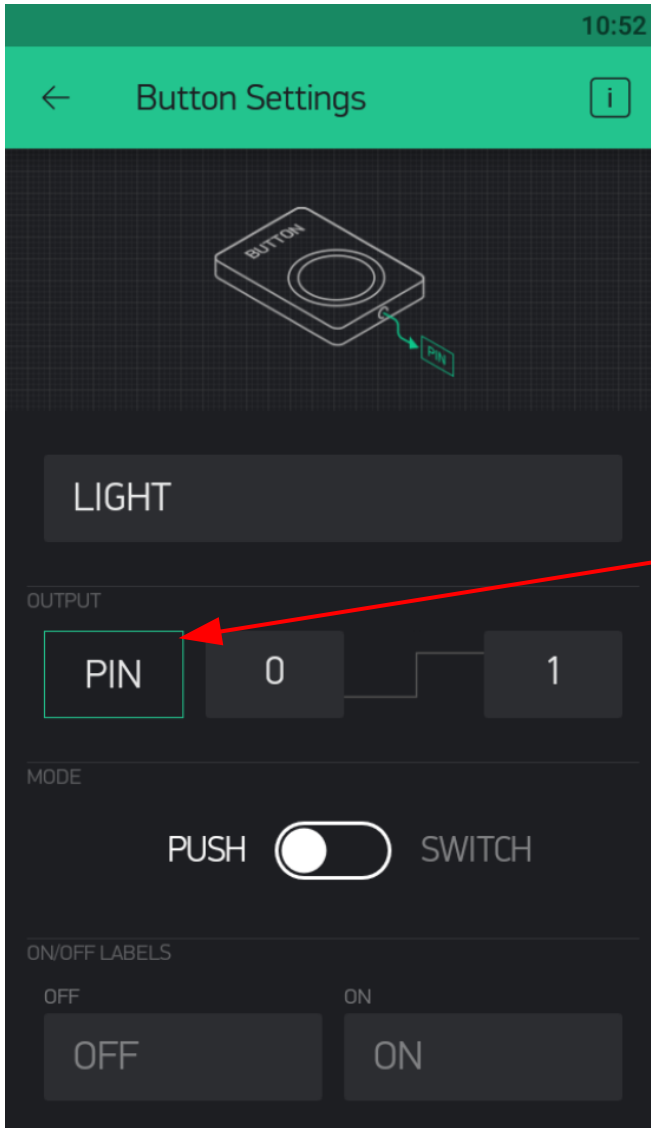
Tap the "+" to open a widget box

Tap this...

...to add a button to your screen



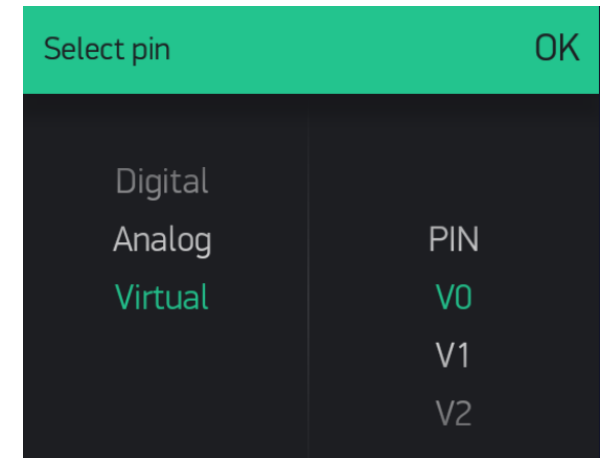
# Challenge 2 : Add Light Switch



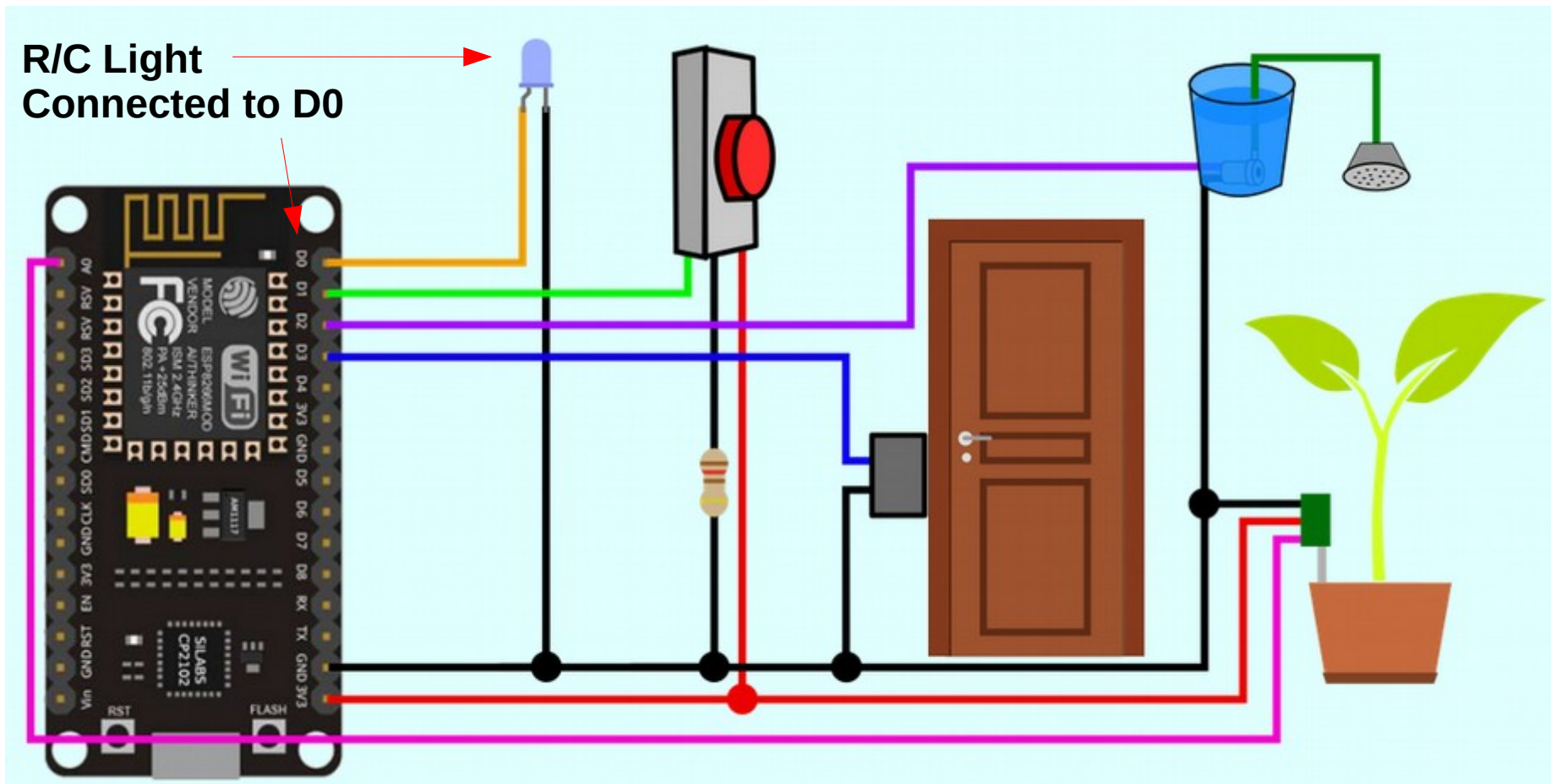
Tap your newly added button to configure it

Tap on "PIN" and select "Virtual | V0"

By default, Blynk will send a 0 when the button is released, and a 1 when the button is pressed. You can modify this by changing the 0 and 1 next to the PIN button.



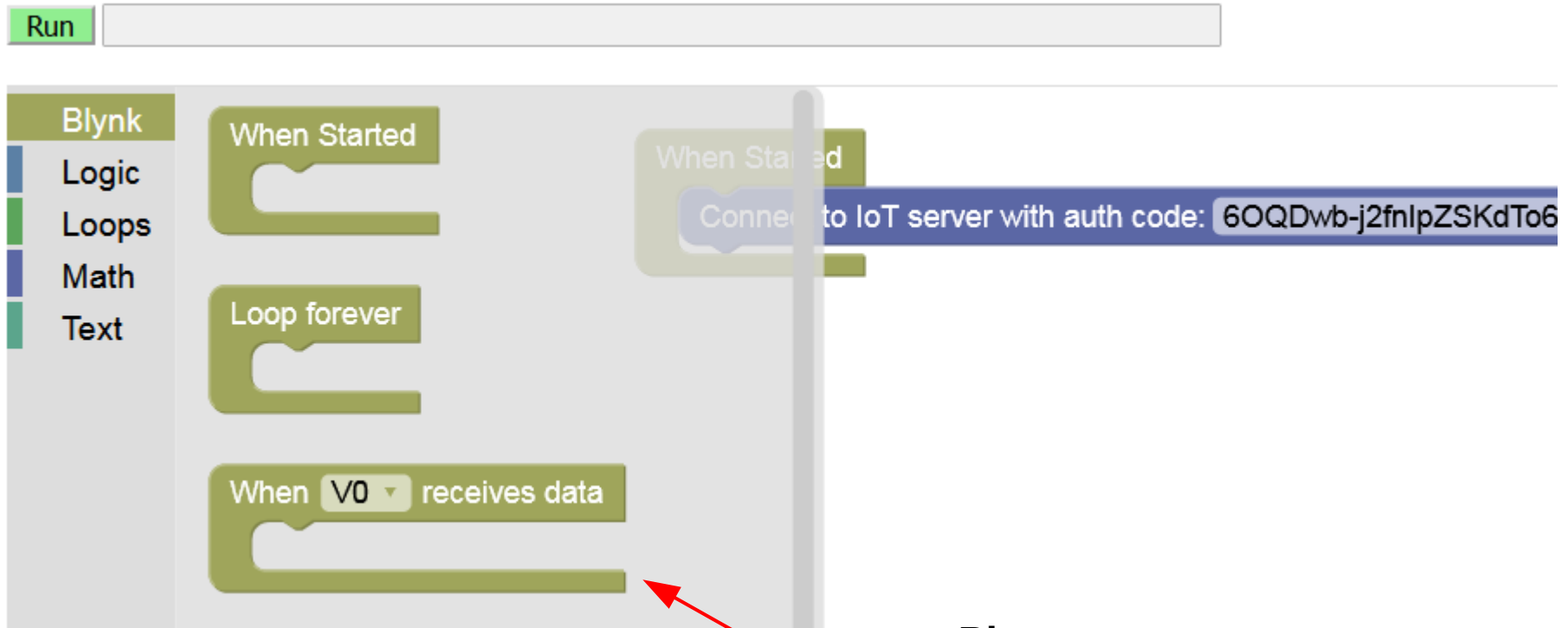
# Challenge 2 : Add Light Switch



<http://a9i.sg/iot>

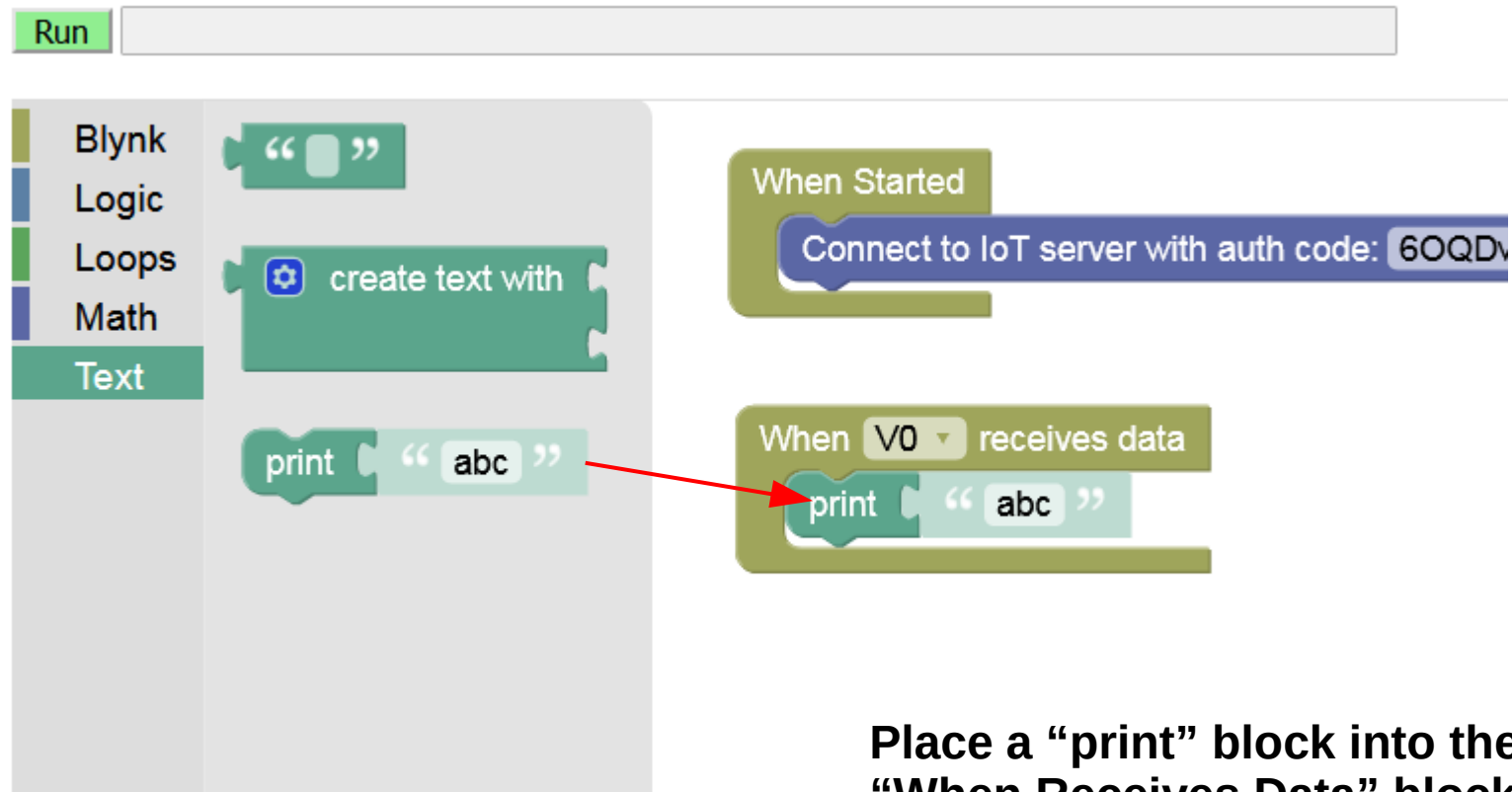
**A POSTERIORI**  
Play · Experience · Learn

# Challenge 2 : Add Light Switch



**Place a  
“When Receives Data”  
block into the your  
programming area**

# Challenge 2 : Add Light Switch



The image shows a code editor interface. At the top, there is a 'Run' button. Below it, a sidebar on the left lists categories: Blynk, Logic, Loops, Math, and Text. The 'Text' category is selected, showing a 'print' block with the text 'abc'. A red arrow points from this block to a 'When V0 receives data' block in the main workspace. The main workspace also contains a 'When Started' block with a 'Connect to IoT server with auth code: 60QDV' block. The 'When V0 receives data' block already has a 'print' block with the text 'abc' inside it.

Place a “print” block into the  
“When Receives Data” block



# Challenge 2 : Add Light Switch

The image shows the Blynk IDE interface. On the left, there is a sidebar with categories: Blynk, Logic, Loops, Math, and Text. Below these are several code blocks: 'When Started', 'Loop forever', 'When V0 receives data', 'Received data', and 'Value of A0'. The 'When V0 receives data' block is selected. In the main workspace, there is a 'When Started' block containing a 'Connect to IoT server with auth code: 6OQDwb-' block, and a 'When V0 receives data' block containing a 'print' block. A red arrow points from the 'Received data' block in the sidebar to the 'print' block in the workspace.

Run

Blynk

Logic

Loops

Math

Text

When Started

Loop forever

When V0 receives data

Received data

Value of A0

When Started

Connect to IoT server with auth code: 6OQDwb-

When V0 receives data

print

Received data

Place a “Received Data” block into the “print” block

Now the program will report whatever value is being sent down from the Blynk Dashboard

# Challenge 2 : Add Light Switch

The screenshot shows the Blynk IDE interface. On the left, a sidebar lists categories: Blynk, Logic, Loops, Math, and Text. The 'Logic' category is selected, showing an 'if' block with a gear icon and a 'do' block. A red arrow points from the gear icon to the 'if' block in the main workspace. The main workspace contains a 'When Started' block with a 'Connect to IoT server with auth code: 6OQDwb-j2fnlpZSKd' block. Below it is a 'When V0 receives data' block containing a 'print Received data' block and an 'if' block with a gear icon. A red arrow points from the gear icon to the 'if' block. A callout box shows an 'if' block with an 'else' block being added to its bottom. Another red arrow points from the 'else' block in the callout to the 'if' block in the main workspace.

Place an “if” block under the “print” block

Click the gear icon and add an “else” under the “if”

# Challenge 2 : Add Light Switch

Run

The screenshot shows the Blynk IDE interface. On the left, there is a sidebar with categories: Blynk, Logic, Loops, Math, and Text. Under the Logic category, there is an 'if' block and a comparison block (two boxes with an equals sign). The main workspace contains a 'When Started' block with a 'Connect to IoT server with auth code: 6OQDwb-j...' block. Below it is a 'When V0 receives data' block. Inside this block, there is a 'print Received data' block, followed by an 'if' block. A red arrow points from the 'if' block in the sidebar to the 'if' block in the workspace. The workspace also shows a comparison block (two boxes with an equals sign) that is being positioned next to the 'if' block.

**Place a comparison block  
Next to the “if”**

# Challenge 2 : Add Light Switch

Run

- Blynk
- Logic
- Loops
- Math
- Text

When Started

Connect to IoT server with auth code: 6OQDwb-j2fnlpZSKdTo67ZvExM9wmMD\_

When V0 receives data

print Received data

if

do

else

Duplicate

Add Comment

Collapse Block

Disable Block

Delete Block

**Duplicate the "Received Data"  
And add to the Comparison block**

# Challenge 2 : Add Light Switch

Run

Blynk  
Logic  
Loops  
Math  
Text

When Started

Connect to IoT server with auth code: 6OQDwb-j2fnlpZSKdTo6

When V0 receives data

print Received data

if Received data = 1

do

else

Add a “Number” or “0” block from to the other side of the comparison, and change it to a “1”

# Challenge 2 : Add Light Switch

Run

Blynk

Logic

Loops

Math

Text

When Started

When V0 receives data

Received data

Value of A0

When D1 is pressed

Connect to IoT server with auth code: 6OQDwb-j2fnl

Send Message

Turn D0 on

When Started

Connect to IoT server with auth code: 6OQDwb-j2fnl

When V0 receives data

print Received data

if Received data = 1

do Turn D0 on

else Turn D0 off

Add a “Turn Pin On” when V0 sends “1”, and a “Turn Pin Off” when V0 sends “0”

Click Run and Test!

# Challenge 3: Add a Remote Door Lock

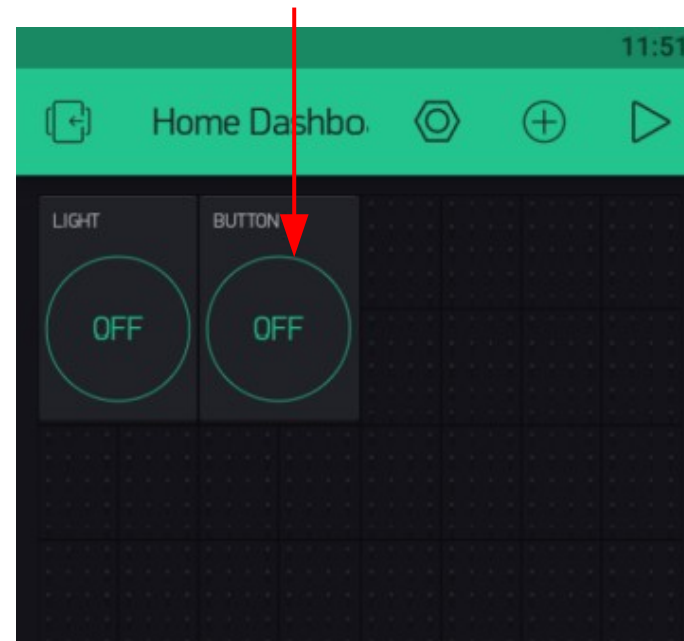
**A POSTERIORI**  
Play · Experience · Learn

# Challenge 3 : Add Door Lock

← Tap the “+” to open a widget box

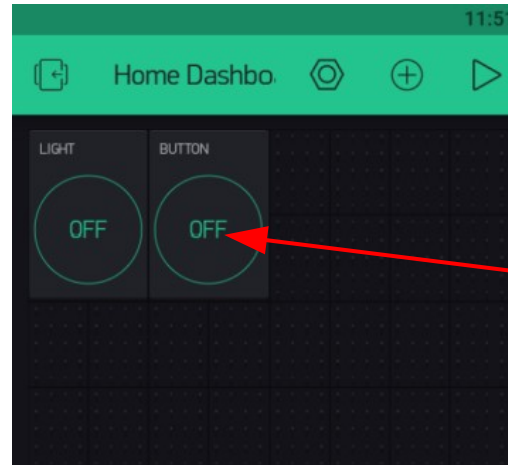
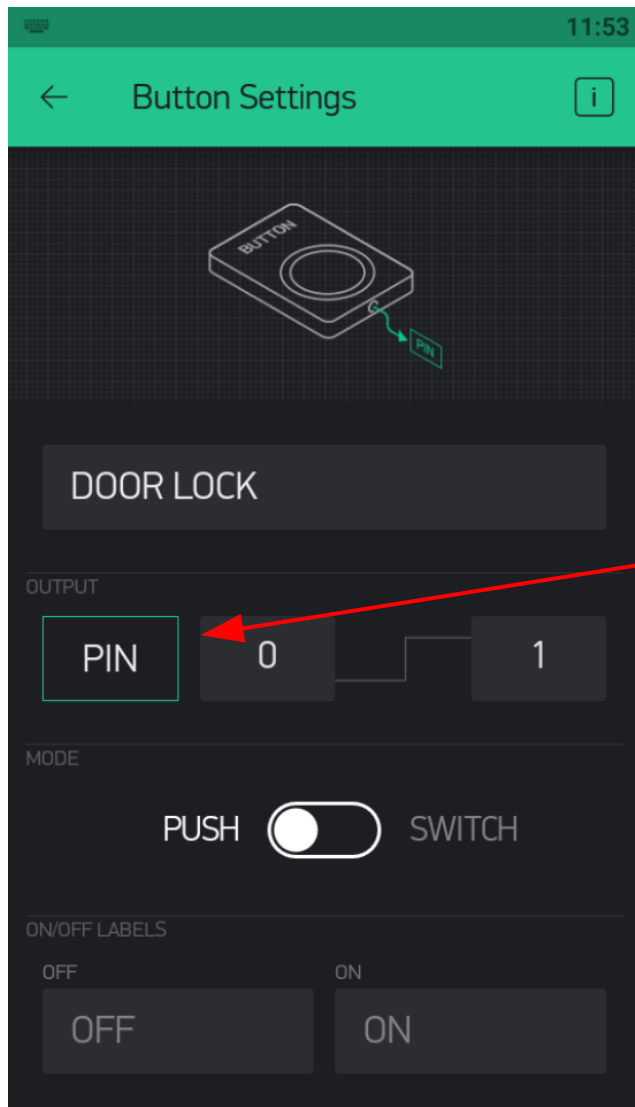
← Tap this...

...to add another button to your screen



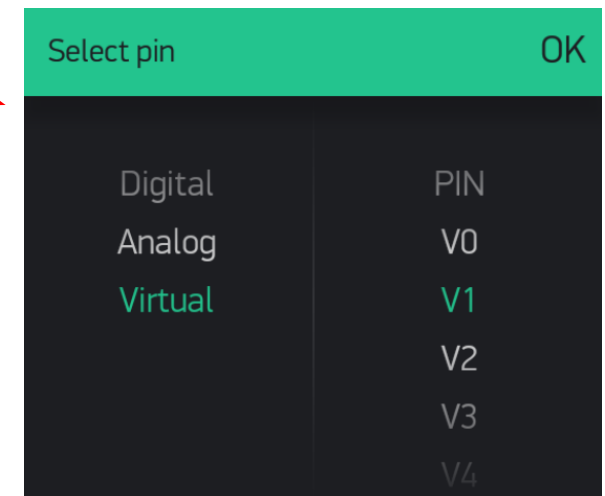


# Challenge 3 : Add Door Lock

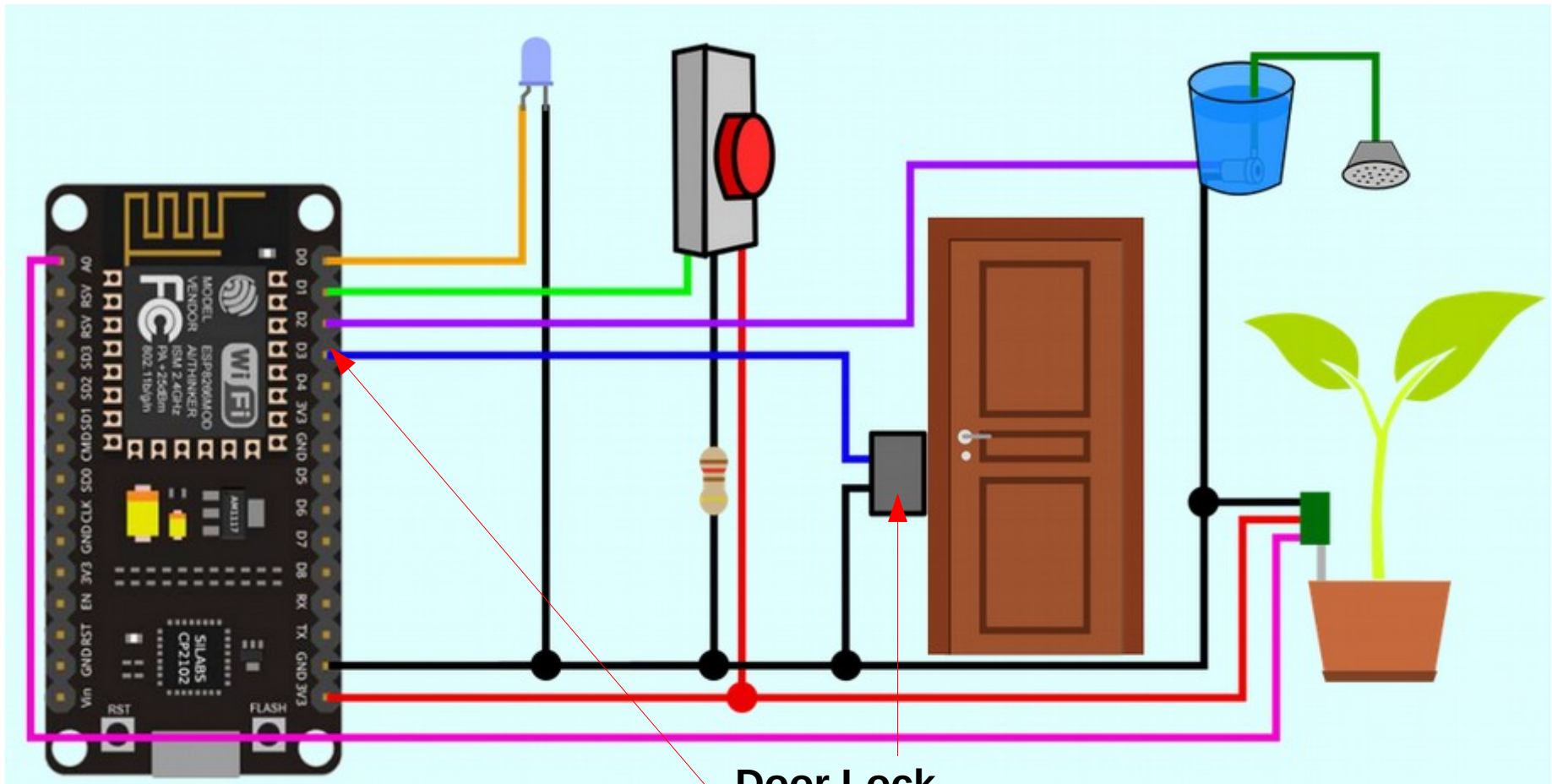


Tap your newly added button to configure it

Tap on "PIN" and select "Virtual | V1"



# Challenge 3 : Add Door Lock



Door Lock  
Connected to D3

<http://a9i.sg/iot>

**A POSTERIORI**  
Play · Experience · Learn

# Challenge 3 : Add Door Lock

Run

Blynk  
Logic  
Loops  
Math  
Text

When Started

Connect to IoT server with auth code: 6OQDwb-j2fnlpZSKdTo67ZvExM9wmMD\_

When V0 receives data

print Received data

if [ ]

do Turn D1 on

else Turn D1 off

Duplicate

Add Comment

Collapse Block

Disable Block

Delete 9 Blocks

When V1 receives data

print Received data

if [ ] = 1

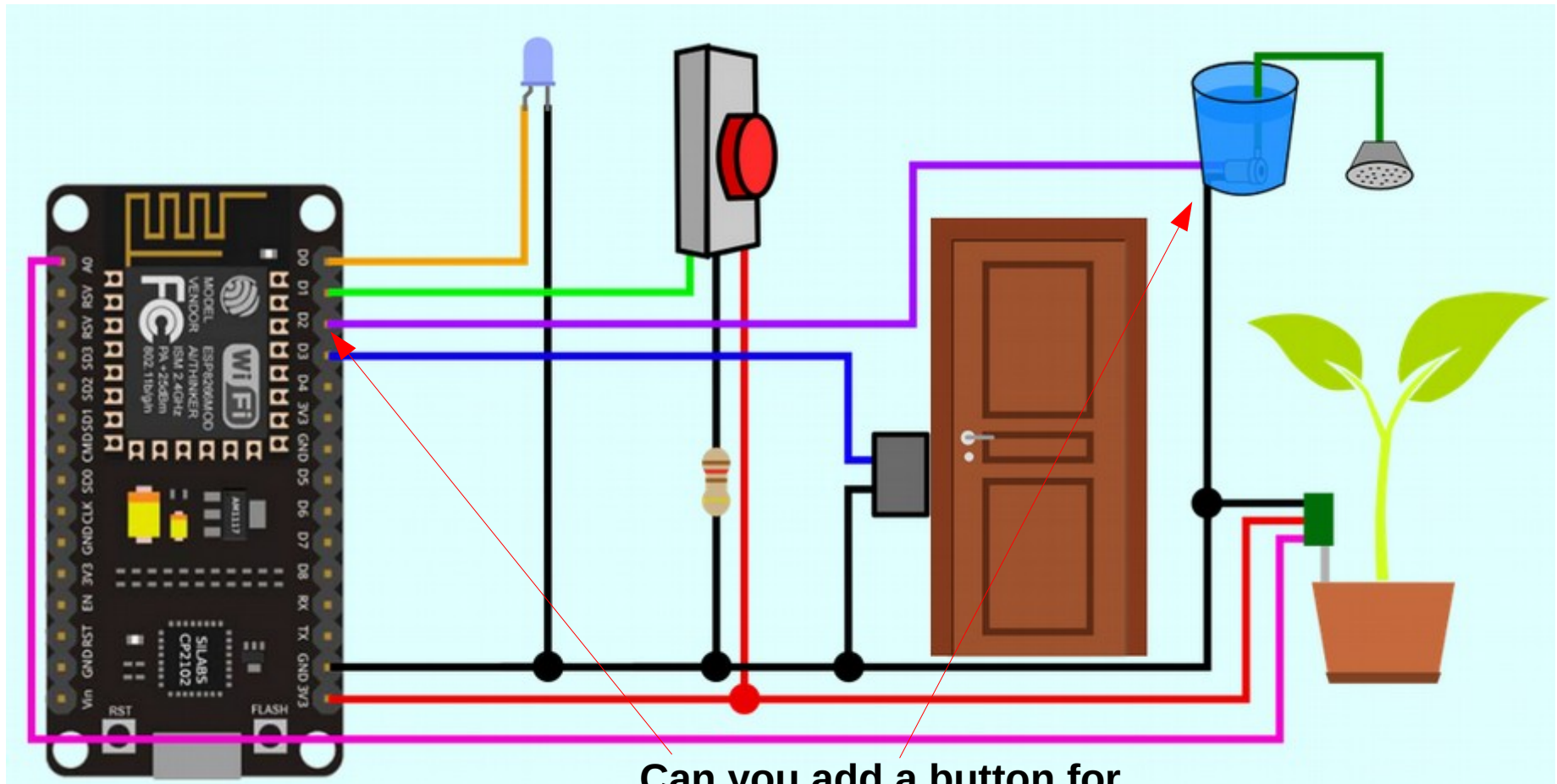
do Turn D3 on

else Turn D3 off

**Duplicate the Light Switch Code**

- Change V0 => V1
- Change D1 => D3

# Challenges and Improvements



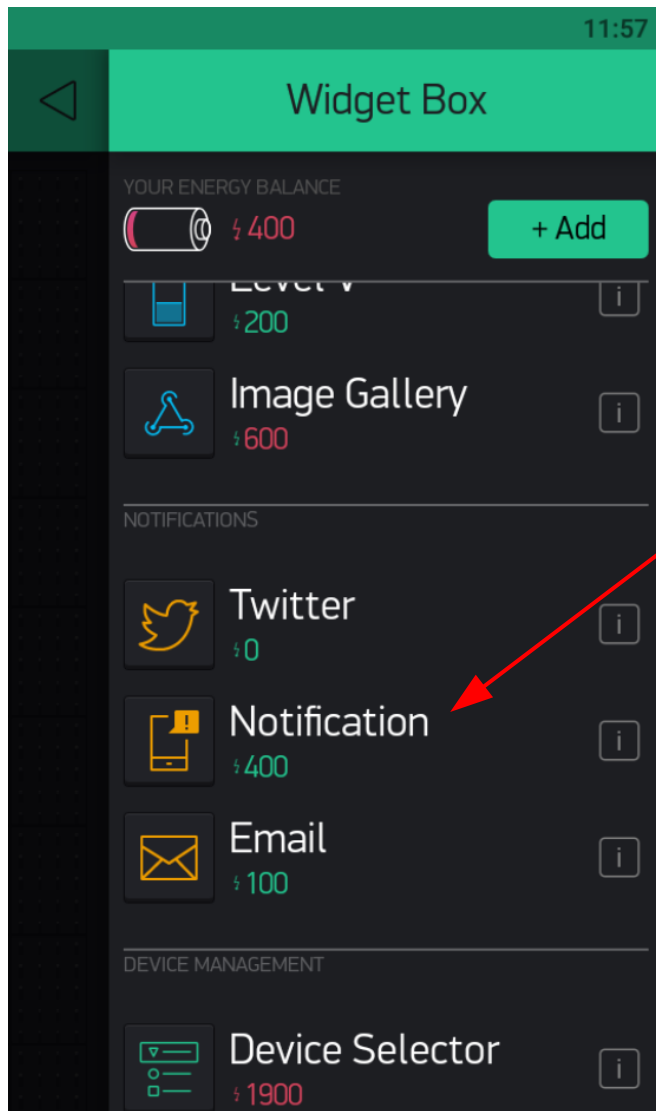
<http://a9i.sg/iot>

**A POSTERIORI**  
Play · Experience · Learn

# Challenge 4: Add a Door Bell Alert

**A POSTERIORI**  
Play · Experience · Learn

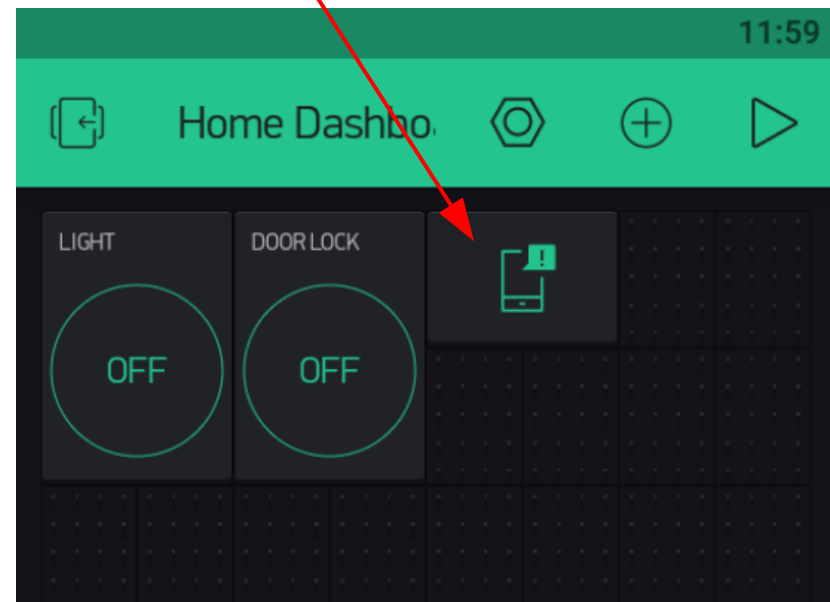
# Challenge 4 : Add Door Bell



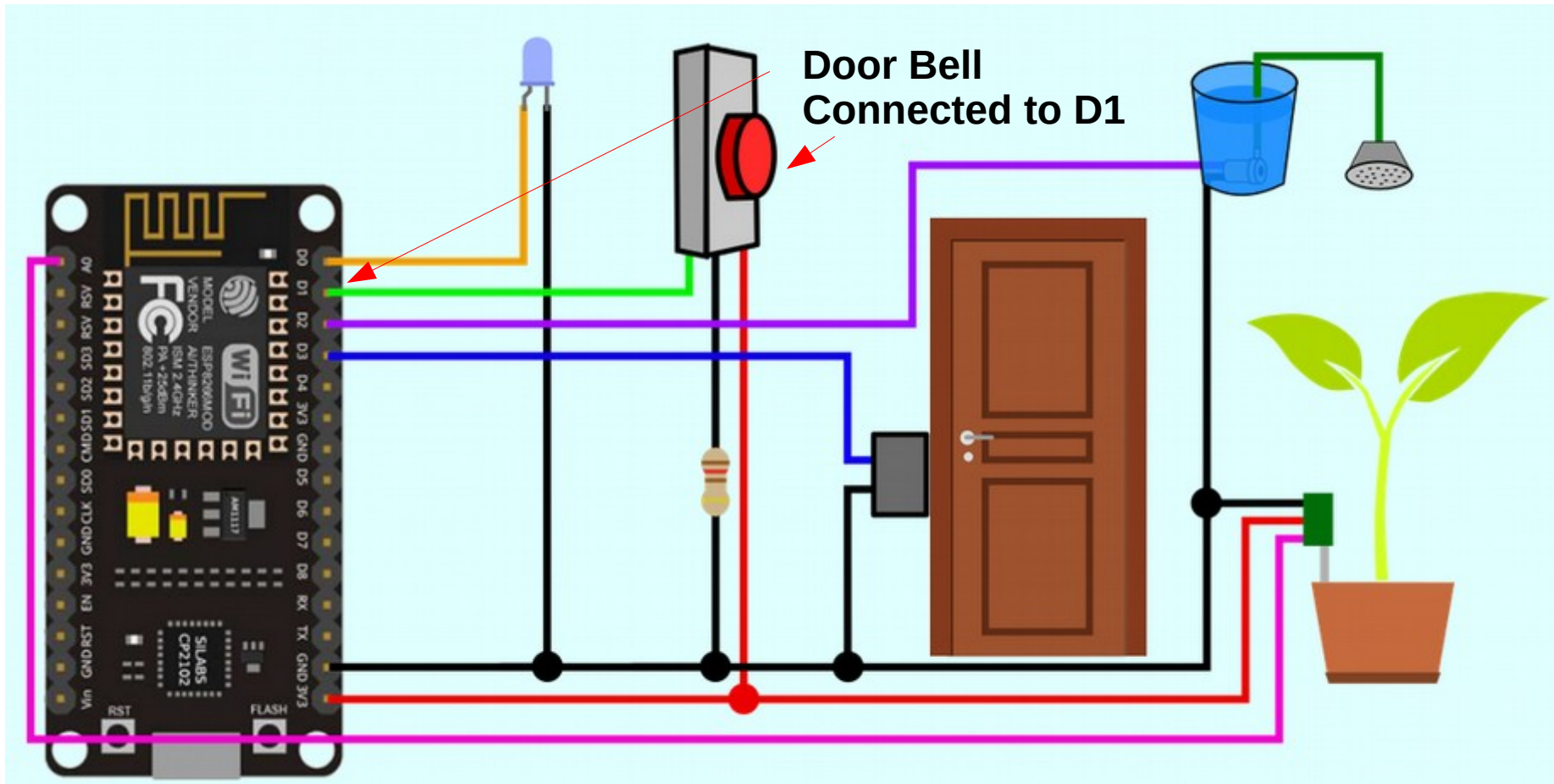
← Tap the “+” to open a widget box

Tap this...

...to add Notification capability



# Challenge 4 : Add Door Bell



<http://a9i.sg/iot>

**A POSTERIORI**  
Play · Experience · Learn

# Challenge 4 : Add Door Bell

Run

Blynk  
Logic  
Loops  
Math  
Text

When Started

Connect to IoT server with auth code: 6OQDwb-j2fnlpZSKdTo67ZvExM9wmMD\_

When D1 is pressed

Send Message " DOOR BELL RINGING! "

Add "When D1 is pressed" block, and insert a "Send Message" block

When V0 receives data

print Received data

if Received data = 1

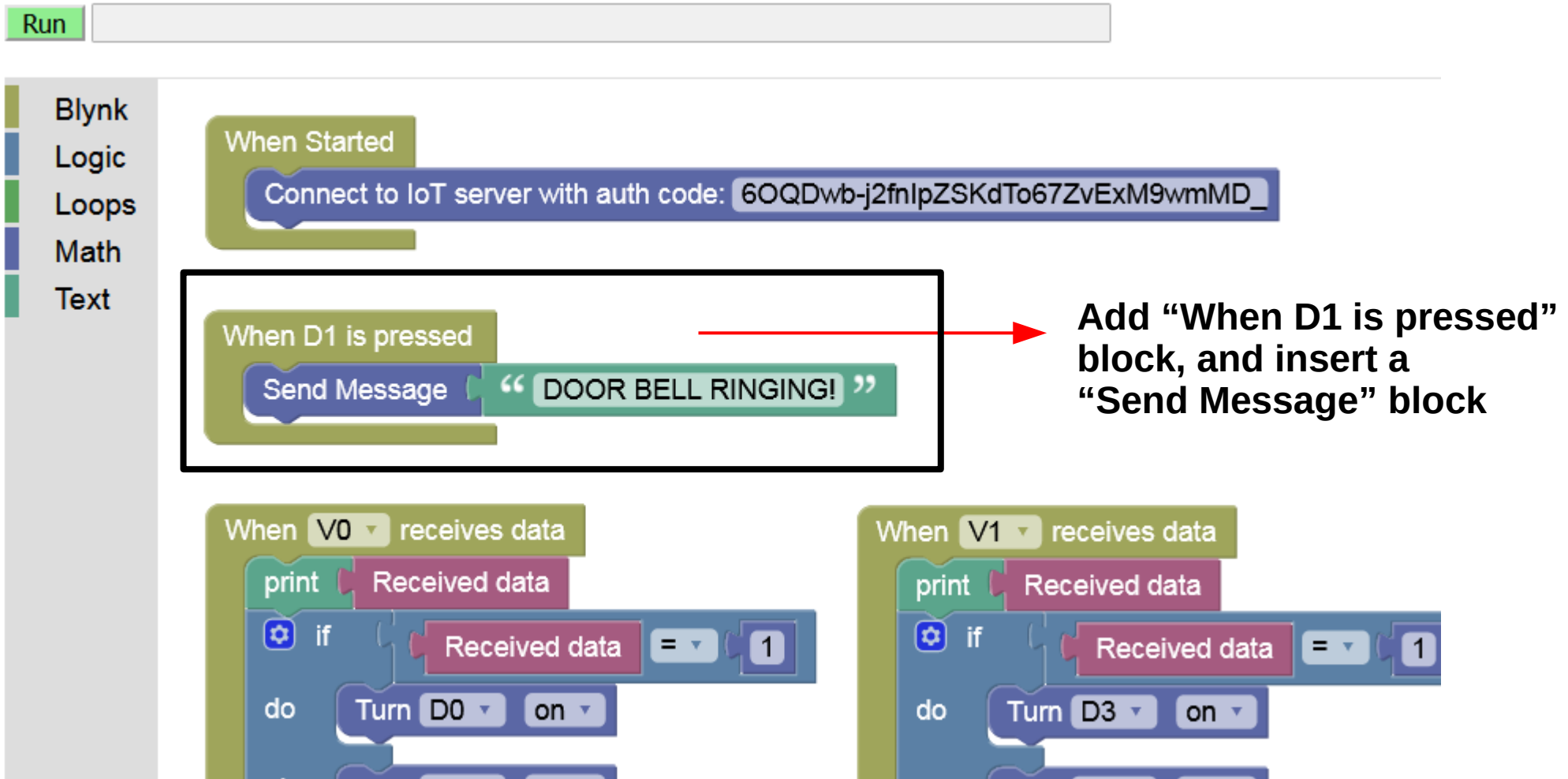
do Turn D0 on

When V1 receives data

print Received data

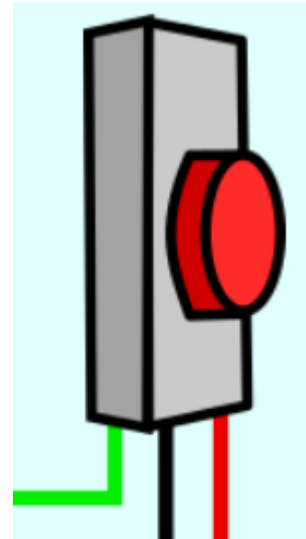
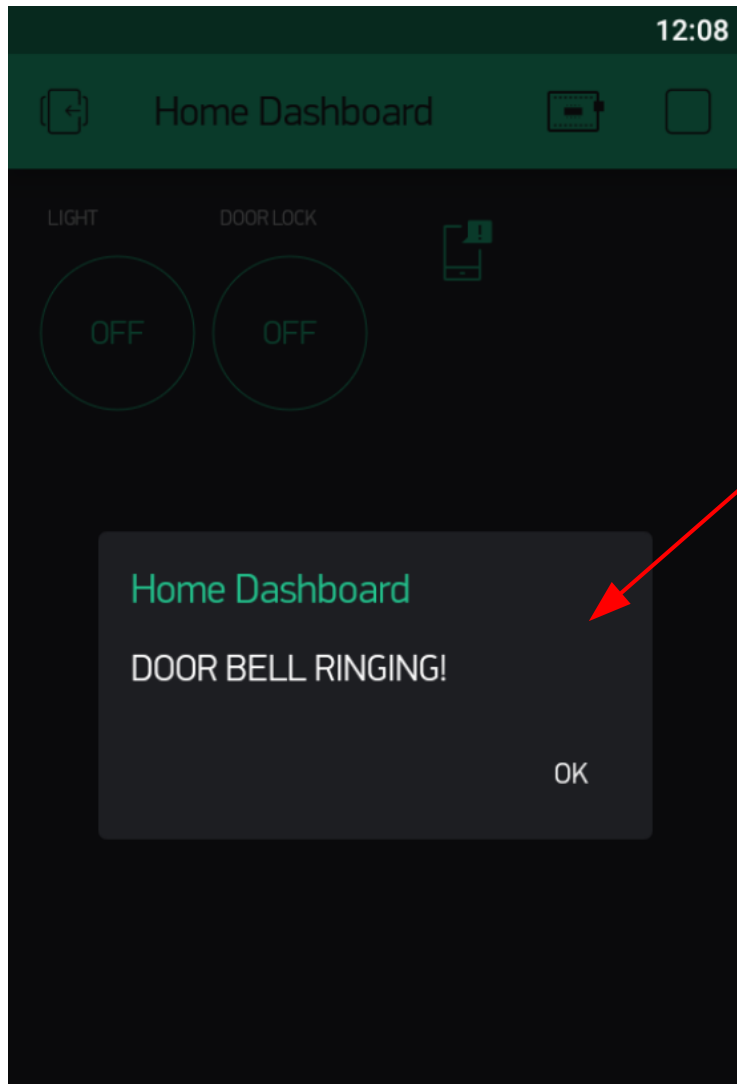
if Received data = 1

do Turn D3 on





# Challenge 4 : Add Door Bell

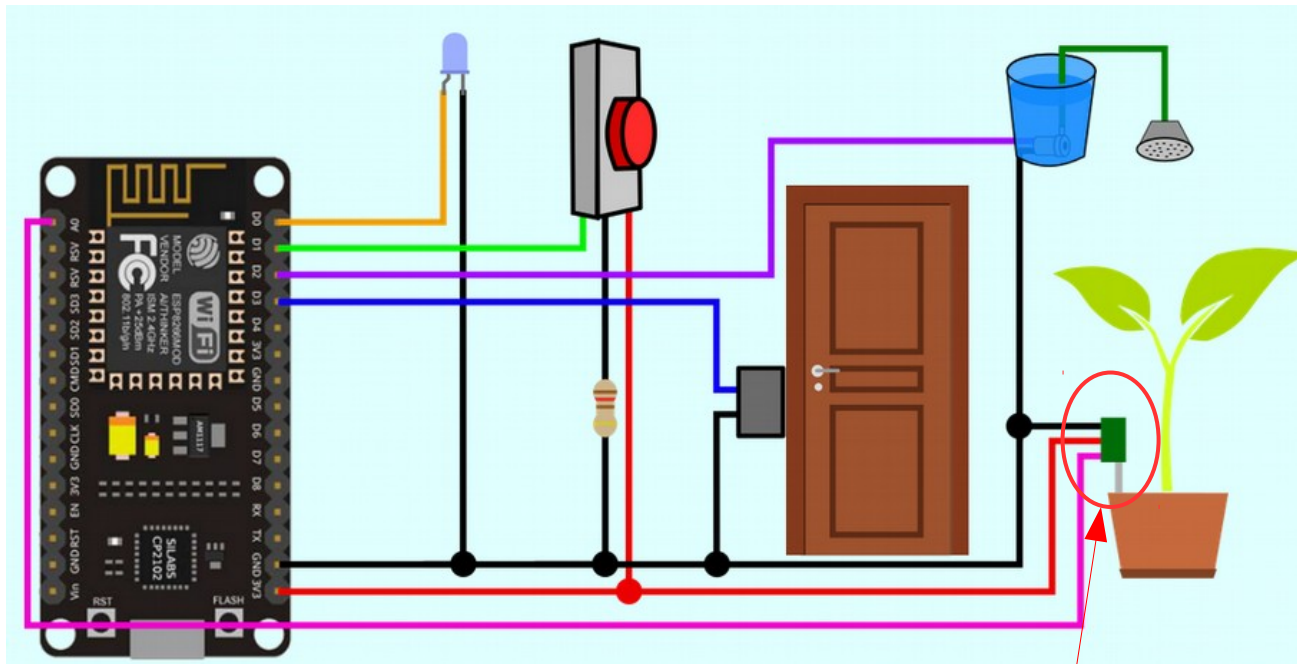


**Click the Door Bell to test the Notification protocol**

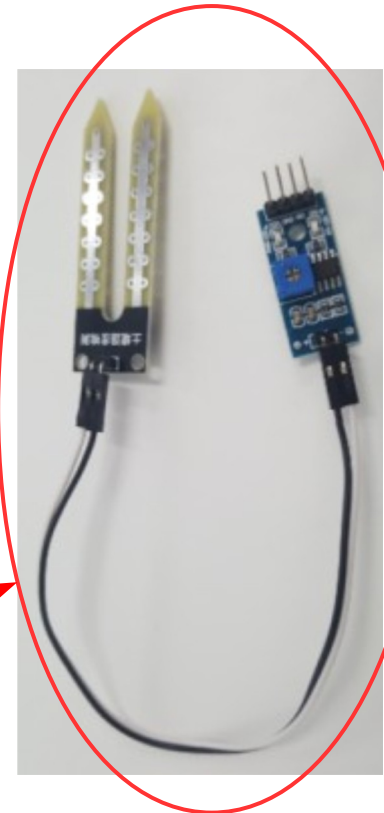
# Challenge 5: Check Soil Moisture

**A POSTERIORI**  
Play · Experience · Learn

# Challenge 5 : Check Soil Moisture

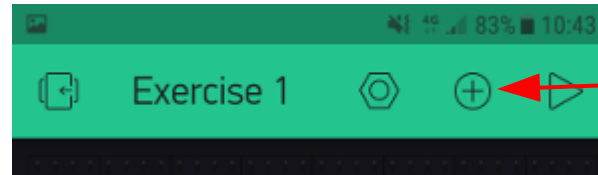
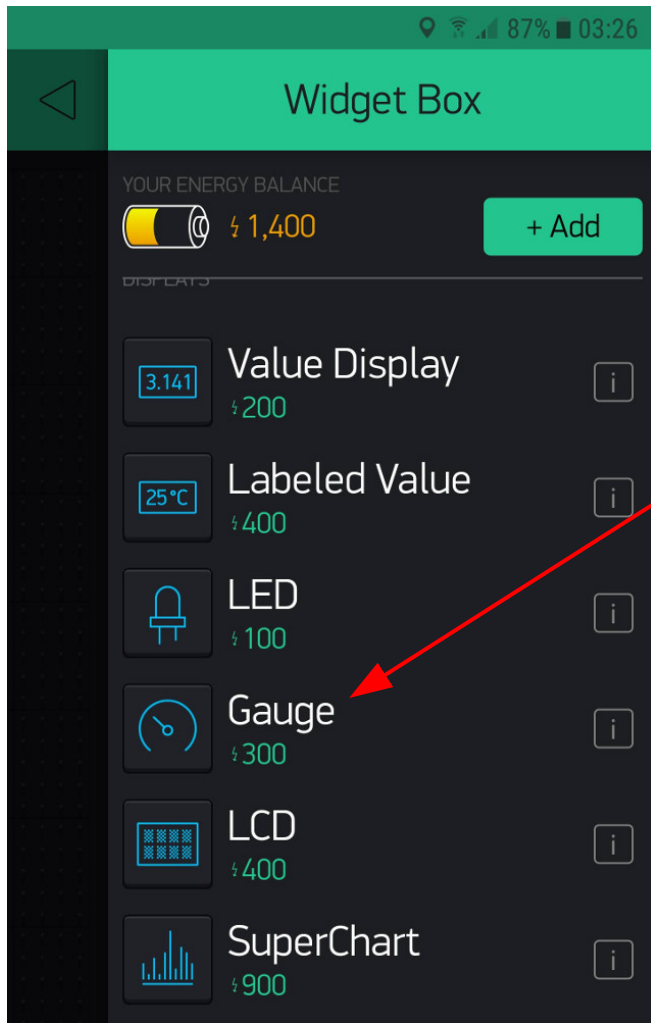


Soil moisture sensor connected to A0



**A POSTERIORI**  
Play · Experience · Learn

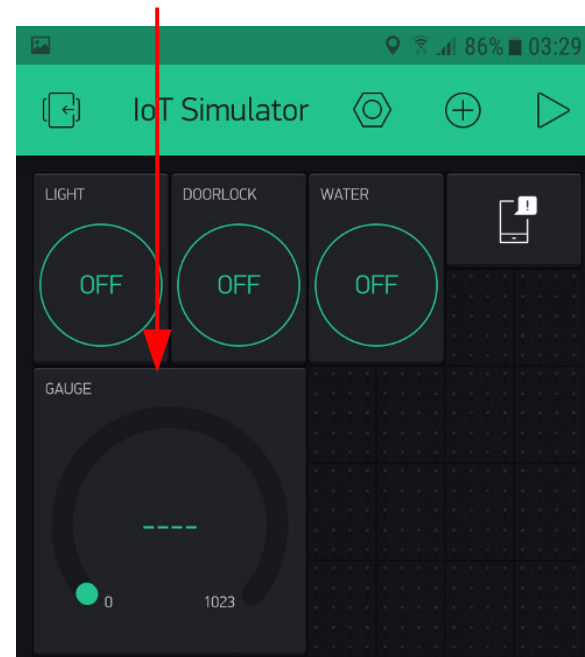
# Challenge 5 : Check Soil Moisture



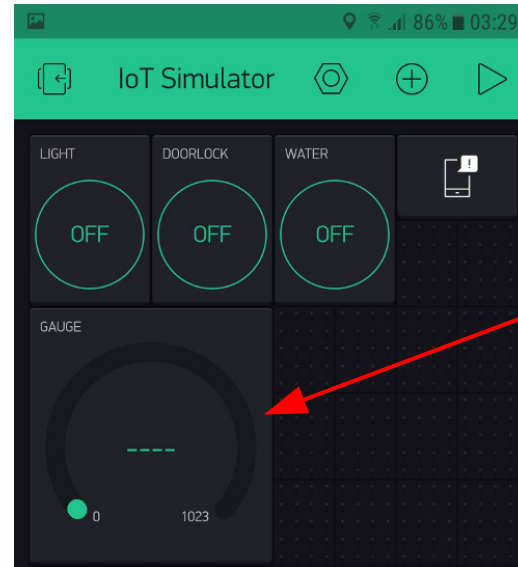
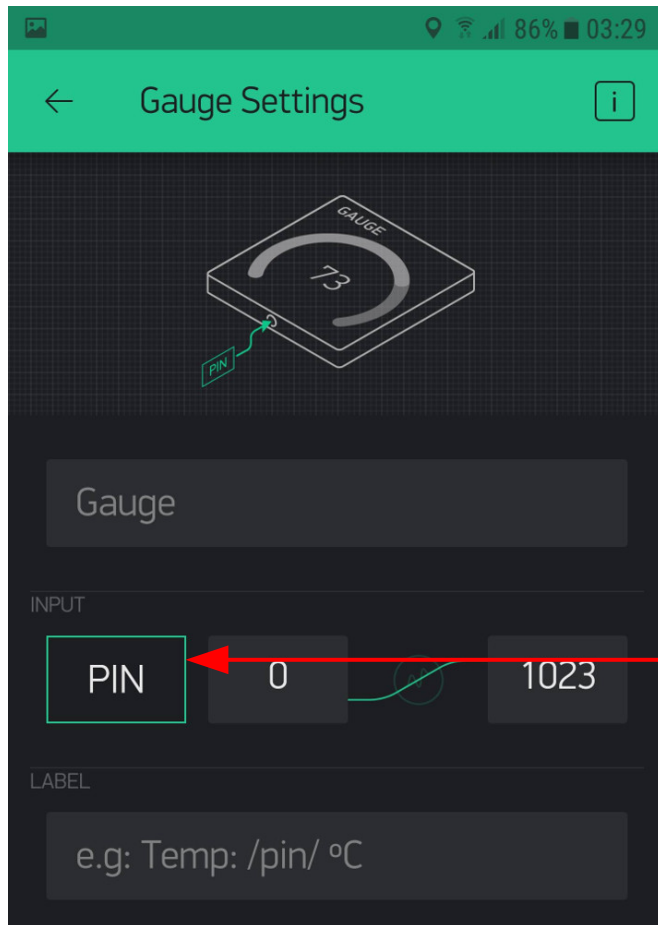
Tap the “+” to open a widget box

Tap this...

...to add a gauge display

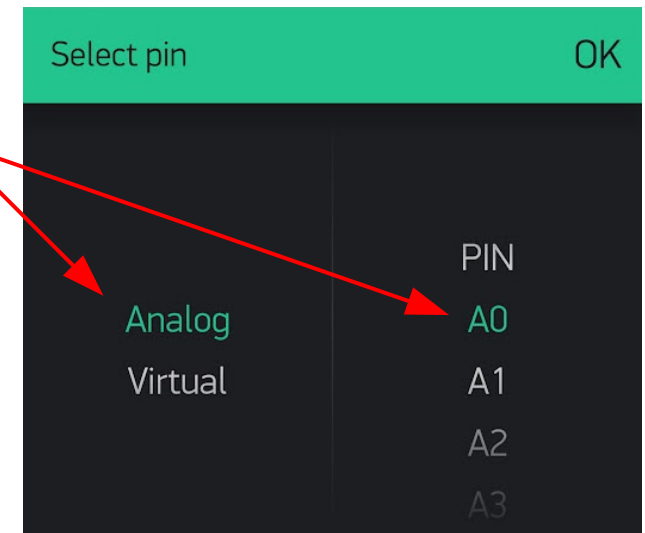


# Challenge 5 : Check Soil Moisture



Tap the gauge to configure it

Tap on "PIN" and select "Analog | A0"



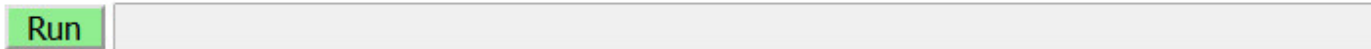
# Challenge 5 : Check Soil Moisture



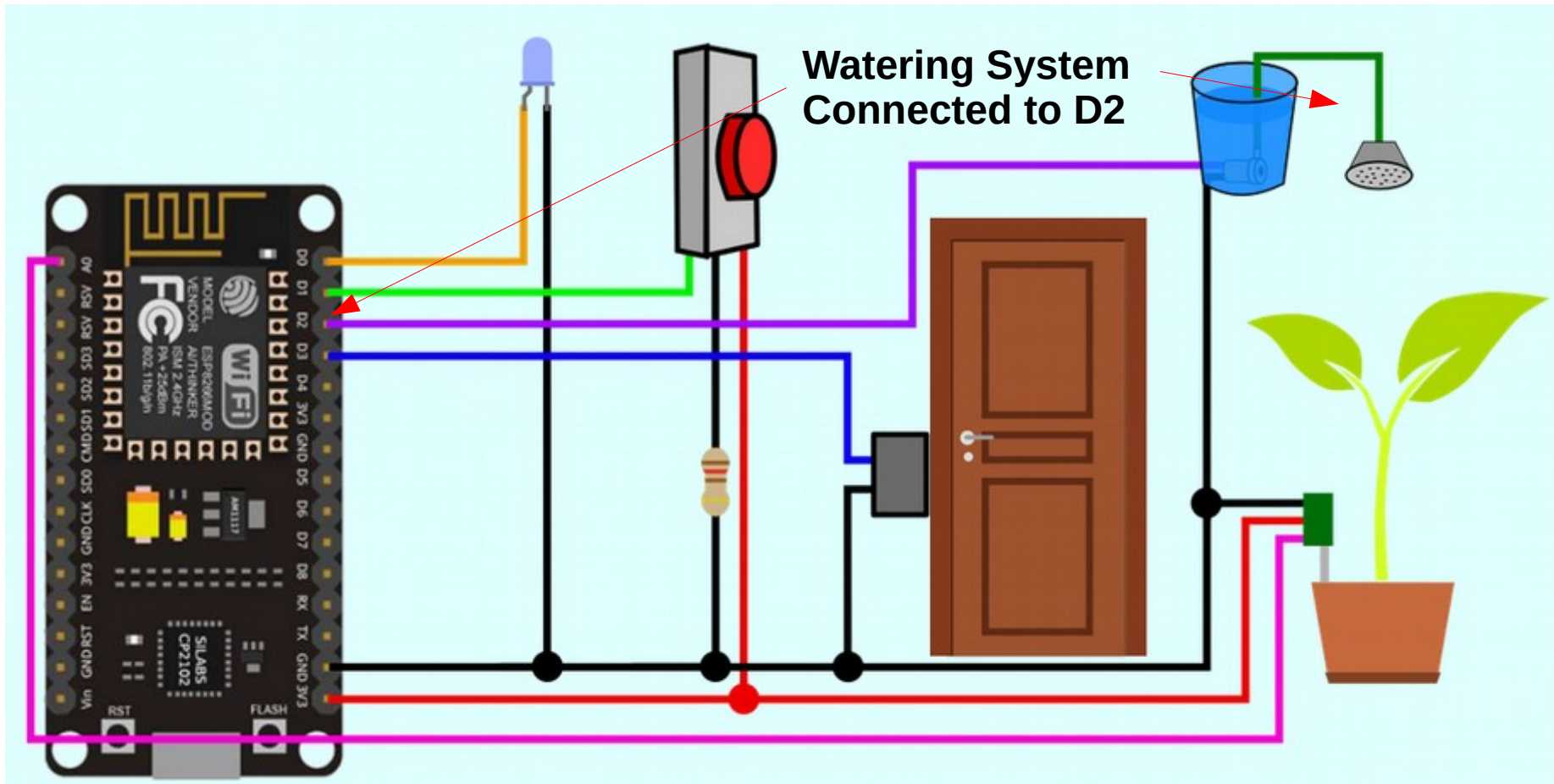
Run both the Blynk app and the IoT simulator

No coding required. Blynk automatically reads from the connected sensor.

You will see the soil moisture level going down.



# Challenge 5b : Add Water



<http://a9i.sg/iot>

**A POSTERIORI**  
Play · Experience · Learn

# Challenges and Improvements

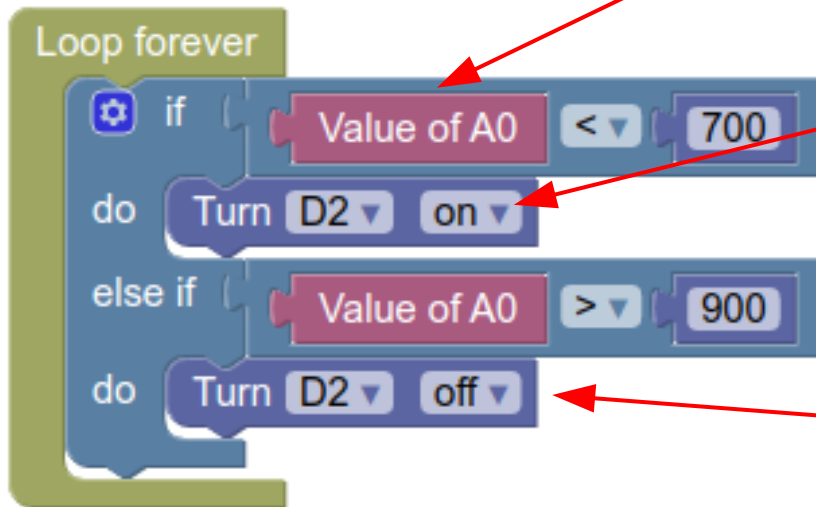
## Automatic Watering

Check if the soil is dry  
(less than 700)

If it is, turn on the watering  
pump

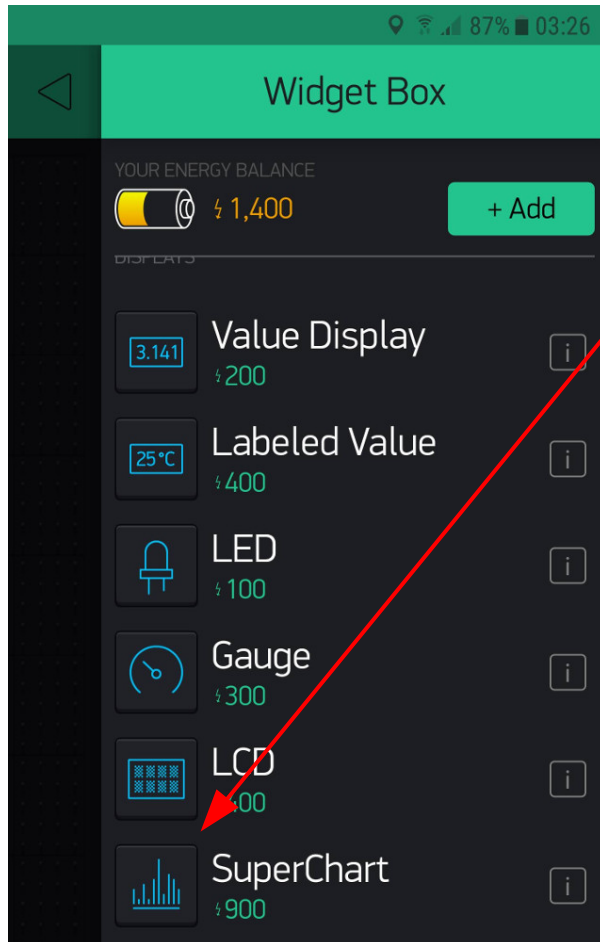
If the soil is moist  
(greater than 900)...

...turn off the watering pump

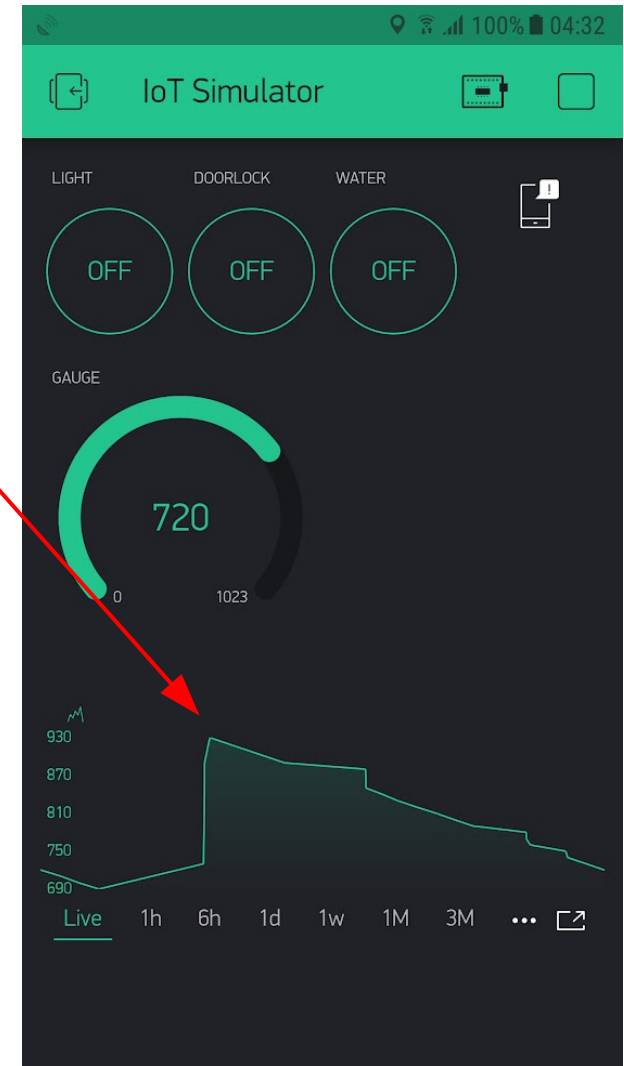




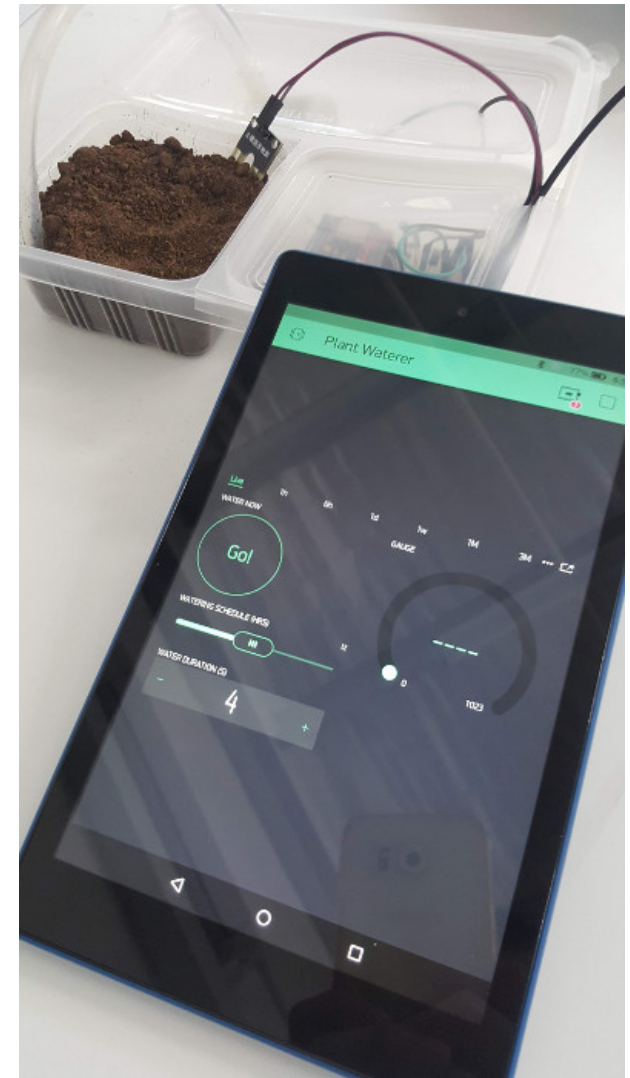
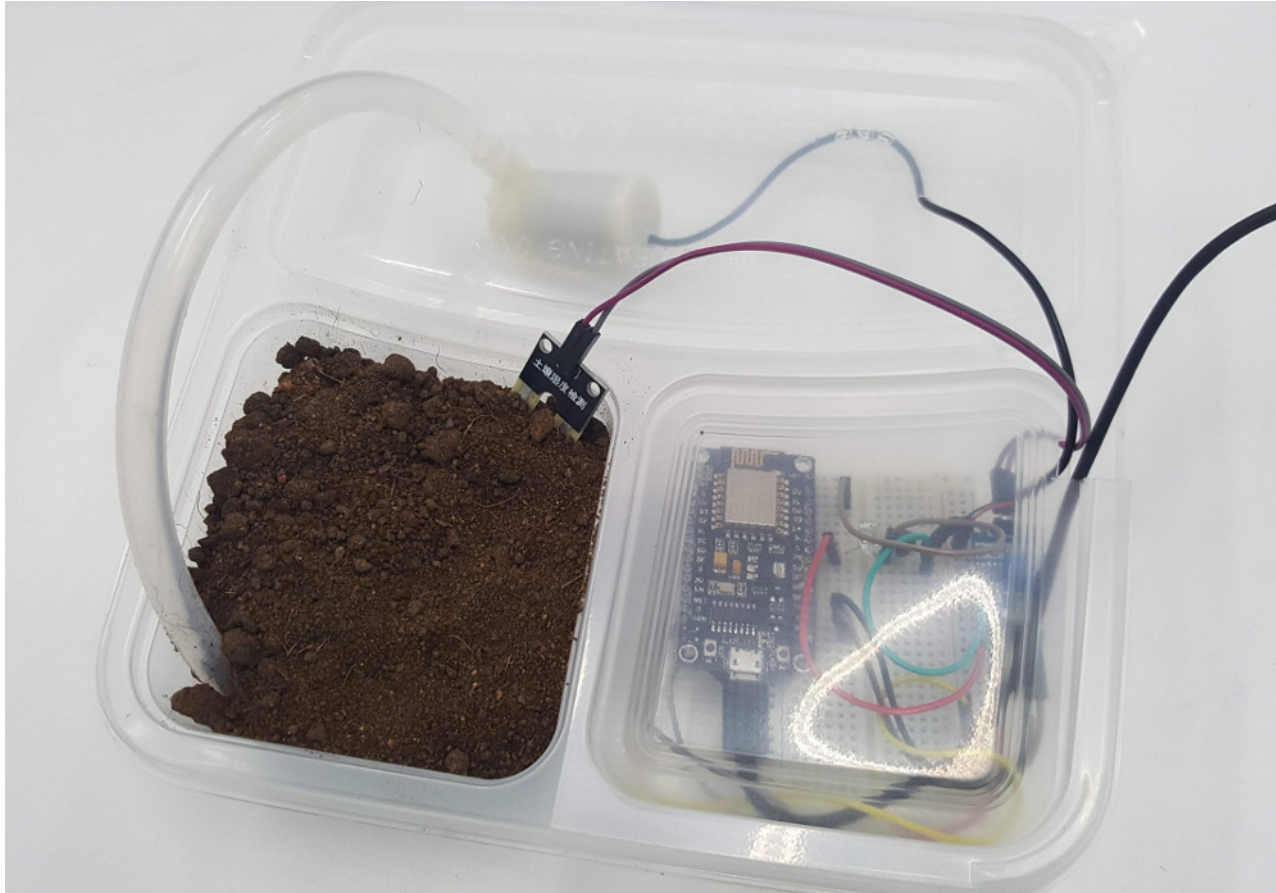
# Challenges and Improvements



**Chart the soil moisture level**



# Challenges and Improvements

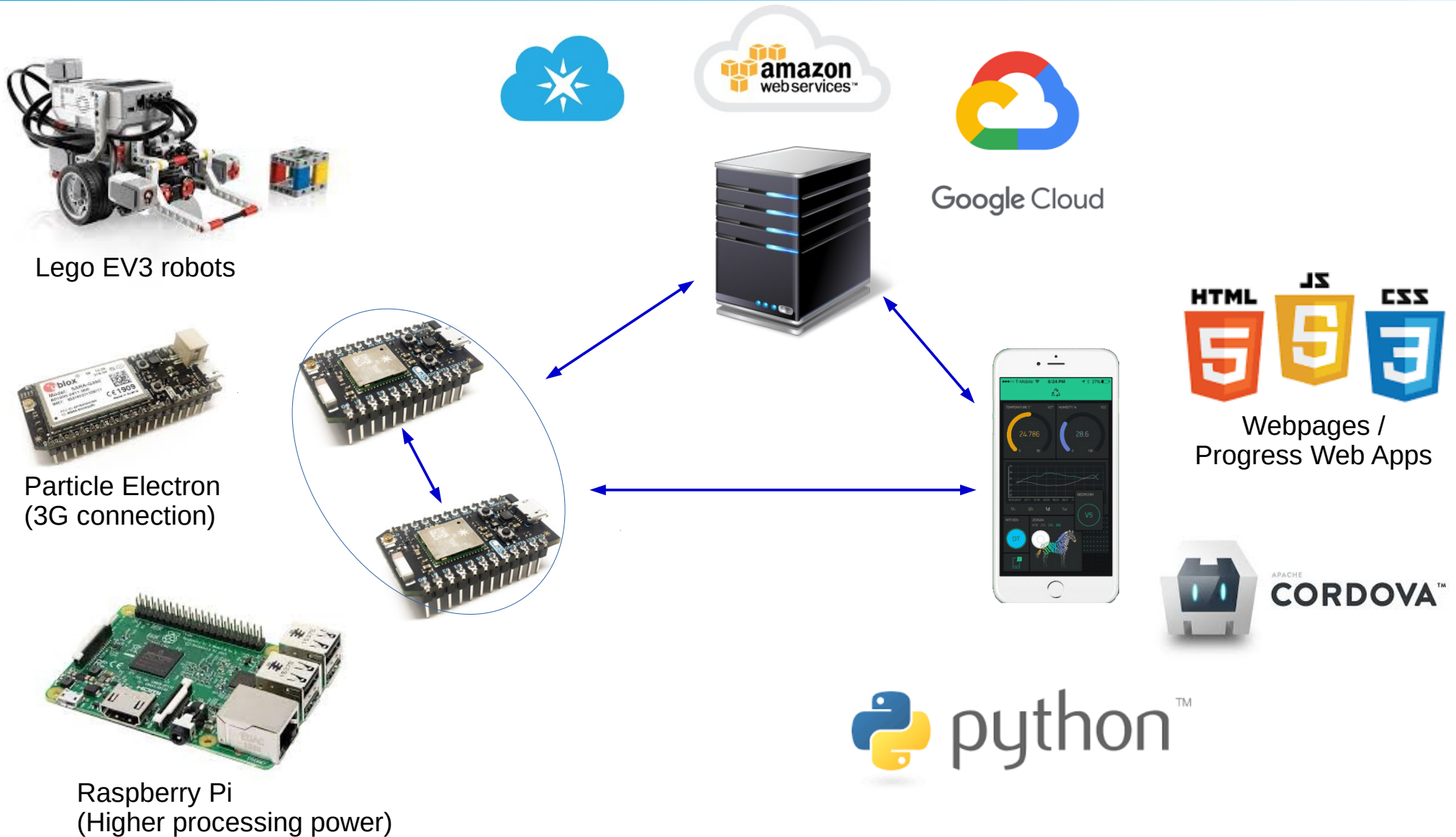


# Challenges and Improvements

## Plant Watering System (complete with controllable LED lights)



# Learning More



# Learning More



LED Strips



Servo Motors



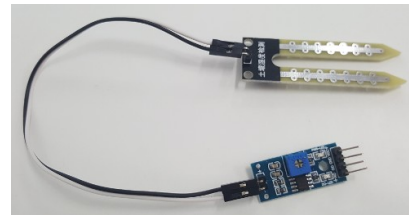
Stepper Motors



RFID Readers



Ultrasonic Distance Sensor



Temperature and Humidity



Camera  
(for image recognition  
or QR Codes)



Wheeled motors  
(for robots)



Gyro and  
acceleration



Laser Distance  
Sensor

# Copyright

- Created by A Posteriori LLP
- Visit <http://aposteriori.com.sg/> for more tips and tutorials
- This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

